

# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



## THESIS

### REDUCING NON-MONOTONICITIES IN COMBAT MODELS

by

William C. Vinyard

September 2001

Thesis Advisor:  
Co-Advisor  
Associate Advisor:

Thomas W. Lucas  
Carlos F. Borges  
Paul J. Sanchez

**Approved for public release; distribution is unlimited**

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
<b>1. AGENCY USE ONLY (Leave blank)</b>		<b>2. REPORT DATE</b> September 2001	<b>3. REPORT TYPE AND DATES COVERED</b> Master's Thesis	
<b>4. TITLE AND SUBTITLE:</b> Title (Mix case letters) Reducing Non-Monotonicities In Combat Models			<b>5. FUNDING NUMBERS</b>	
<b>6. AUTHOR(S)</b> Vinyard, W. C.				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Naval Postgraduate School Monterey, CA 93943-5000			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> N/A			<b>10. SPONSORING / MONITORING AGENCY REPORT NUMBER</b>	
<b>11. SUPPLEMENTARY NOTES</b> The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited			<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT</b>  <p>Non-monotonic behavior in combat models is an important topic to those using the output of such models as a basis for decision making. These decisions can be complicated by non-monotonic behavior in the combat models. This paper examines the Dewar model which exhibits non-monotonic behavior caused by the chaos inherent in its structure. Previous papers have examined only small subsets of this 18 dimensional combat model. The combinatorial possibilities of main effects and interactions among the 18 dimensions are too great to examine en masse. Consequently, we have three goals. First, systematically explore the Dewar model for additional non-monotonic behavior. Second, determine the effect of stochastic modeling on the non-monotonic behavior of the Dewar model response surface. Third, we develop a method for measuring non-monotonicity in the response surface generated by the model. Latin Hypercube Sampling discovers non-monotonicity across broad regions of the model's phase space, and in multiple measures of effectiveness. Stochastic perturbation of model parameters has a dramatic effect on the non-monotonicity of the response surface. Stochastic perturbation can both reduce and exacerbate the non-monotonic behavior of the response surface. If done properly, stochastic modeling can significantly improve the interpretability of the response surface.</p>				
<b>14. SUBJECT TERMS</b> Modeling and Simulation, Stochastic Modeling, Combat Models, Fractional Factorial Experiments, Latin Hypercube Sampling, Response Surfaces, Chaos, Strange Attractors, Fractals, Java, Non-monotonicity.			<b>15. NUMBER OF PAGES</b> 139	
			<b>16. PRICE CODE</b>	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b> UL	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**REDUCING NON-MONOTONICITIES IN COMBAT MODELS**

William C. Vinyard  
Major, United States Marine Corps  
B.A., University of Colorado, Boulder, 1988

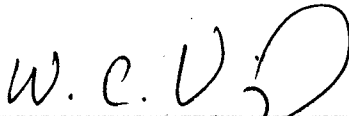
Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN OPERATIONS RESEARCH**


from the

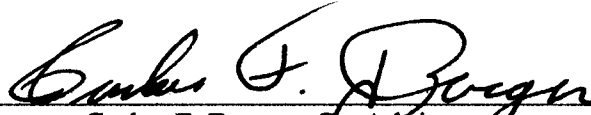
**NAVAL POSTGRADUATE SCHOOL  
September 2001**

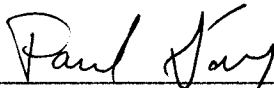
Author:

  
William C. Vinyard

Approved by:

  
Thomas W. Lucas, Thesis Advisor

  
Carlos F. Borges, Co-Advisor



Paul J. Sanchez, Associate Advisor



James N. Eagle, Chairman  
Operations Research Department

THIS PAGE INTENTIONALLY LEFT BLANK

## **ABSTRACT**

Non-monotonic behavior in combat models is an important topic to those using the output of such models as a basis for decision making. These decisions can be complicated by non-monotonic behavior in the combat models. This paper examines the Dewar model which exhibits non-monotonic behavior caused by the chaos inherent in its structure. Previous papers have examined only small subsets of this 18 dimensional combat model. The combinatorial possibilities of main effects and interactions among the 18 dimensions are too great to examine en masse. Consequently, we have three goals. First, systematically explore the Dewar model for additional non-monotonic behavior. Second, determine the effect of stochastic modeling on the non-monotonic behavior of the Dewar model response surface. Third, we develop a method for measuring non-monotonicity in the response surface generated by the model. Latin Hypercube Sampling discovers non-monotonicity across broad regions of the model's phase space, and in multiple measures of effectiveness. Stochastic perturbation of model parameters has a dramatic effect on the non-monotonicity of the response surface. Stochastic perturbation can both reduce and exacerbate the non-monotonic behavior of the response surface. If done properly, stochastic modeling can significantly improve the interpretability of the response surface.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

<b>I.</b>	<b>INTRODUCTION.....</b>	<b>1</b>
<b>A.</b>	<b>PREAMBLE.....</b>	<b>1</b>
<b>B.</b>	<b>CHAOS AND NON-MONOTONICITY IN COMBAT MODELS .....</b>	<b>2</b>
<b>C.</b>	<b>PREVIOUS RESEARCH PERTINENT TO THIS STUDY .....</b>	<b>5</b>
<b>D.</b>	<b>METHOD .....</b>	<b>9</b>
1.	Java, Numerics and the Effect of Finite Precision on the Battle Trace.....	9
2.	Latin Hypercube Sampling .....	9
3.	Fractional Factorial Experiment .....	10
4.	Defining Non-monotonic Parameters.....	10
5.	Measures of Effectiveness.....	10
<b>II.</b>	<b>RECODING THE MODEL IN JAVA.....</b>	<b>13</b>
<b>A.</b>	<b>A BRIEF DESCRIPTION OF THE ORIGINAL DEWAR MODEL .....</b>	<b>13</b>
<b>B.</b>	<b>A STOCHASTIC MODIFICATION AND THE POTENTIAL FOR ADDITIONAL PARAMETERS.....</b>	<b>14</b>
<b>C.</b>	<b>THE RESPONSE SURFACE .....</b>	<b>15</b>
1.	A Description of the Deterministic Response Surface .....	15
2.	A Description of the Stochastic Response Surface.....	17
<b>D.</b>	<b>CHECKING THE NUMERICS OF THE MODEL .....</b>	<b>19</b>
1.	Decision Tables and Structural Defects in Decision Logic.....	19
2.	Computer Arithmetic .....	19
3.	Testing the Effect of Rounding Error and Finite Precision on the Output of the Model .....	20
4.	Results of the Simple Experiment .....	22
5.	The Shadowing Lemma.....	23
<b>III.</b>	<b>LATIN HYPERCUBE SAMPLING: SEARCHING FOR NON-MONOTONICITY .....</b>	<b>29</b>
<b>A.</b>	<b>LATIN HYPERCUBE SAMPLING DESIGN .....</b>	<b>29</b>
1.	Implementing the Latin Hypercube Design .....	30
2.	Modifying the LHS Design.....	30
3.	How Thick Should the Hyperplanes Be? .....	31
4.	Results of the Latin Hypercube Design Experiment .....	33
5.	Non-monotonicity in a Different Measure of Performance .....	33
6.	Can Non-monotonicity Be Reduced or Eliminated? .....	35
<b>IV.</b>	<b>THE FRACTIONAL FACTORIAL EXPERIMENT .....</b>	<b>37</b>
<b>A.</b>	<b>DETERMINING THE EFFECTS OF STOCHASTIC MODELING ON THE MODEL'S RESPONSE SURFACES .....</b>	<b>37</b>
<b>V.</b>	<b>DEFINING NON-MONOTONICITY PARAMETERS .....</b>	<b>39</b>
<b>A.</b>	<b>THE NEED FOR PARAMETERS TO MEASURE NON-MONOTONICITY .....</b>	<b>39</b>



B.	THE NON-MONOTONIC PARAMETER FUNCTION.....	45
C.	TRANSFORMING THE Z-VALUES.....	48
D.	THE EMPIRICAL DISTRIBUTION OF THE NMPF VALUES .....	51
E.	ANALYSIS OF NMPF VALUES WITH RESPECT TO $\lambda$ , POINT DENSITY, NUMBER OF REPLICATIONS AND TYPE OF RESPONSE SURFACE.....	62
F.	RESULTS OF FULL FACTORIAL DESIGN EXPERIMENT.....	64
1.	ANOVA Results for NmP.....	64
2.	ANOVA Results for Jumps (J) .....	64
3.	ANOVA Results for Significant Jumps (SJ).....	65
4.	ANOVA Results for NmT .....	66
5.	ANOVA Results for Trends (T).....	67
6.	ANOVA Results for Significant Trends (ST) .....	68
G.	APPLYING THE NMPF TO ACTUAL DATA.....	68
H.	CONCLUDING REMARKS REGARDING THE NMPF .....	76
VI.	FRACTIONAL FACTORIAL EXPERIMENT RESULTS .....	79
A.	ANALYSIS .....	79
1.	Correlation Among the Responses and the Method of Analysis ..	79
2.	Graphical Diagnostics: Confirming Correlation .....	80
3.	Describing the Landscape of the Stochastic Response Surface .....	81
4.	Explaining the Negative Correlation.....	82
5.	Classifying a Stochastic Response Surface .....	83
6.	Fitting the Model.....	84
7.	Calculating the Effect of the Factors on the Response Variables..	85
8.	A Simple Integer Program to Minimize Non-monotonicity.....	87
9.	Testing the Analysis .....	89
B.	FINAL COMMENTS ON THE RESULTS OF THE FRACTIONAL FACTORIAL EXPERIMENT .....	92
VII.	CONCLUSIONS .....	95
A.	PRIMARY FINDINGS.....	95
B.	SUGGESTED FOLLOW-ON RESEARCH .....	97
APPENDIX A.	SPLUS CODE FOR THE NMPF .....	99
APPENDIX B.	NMPF EXPERIMENT RESULTS AND ANALYSIS.....	105
A.	RESULTS OF EXPERIMENT.....	105
APPENDIX C.	FRACTIONAL FACTORIAL RESULTS .....	107
LIST OF REFERENCES	.....	109
INITIAL DISTRIBUTION LIST	.....	111

## LIST OF FIGURES

Figure 0.1.	The Response Surface of the Original Dewar Model. ....	xix
Figure 0.2.	Two Dimensional Subspaces Exhibiting Non-monotonic Behavior. ....	xxii
Figure 0.3.	Careful Stochastic Modeling Can Smooth Non-monotonic Trends. ....	xxiv
Figure 1.1.	A Trait of Chaos: Sensitivity to Initial Conditions. ....	3
Figure 2.1.	Binary Response Surface of Original Dewar Model. ....	16
Figure 2.2.	An Example of a Stochastic Response Surface. ....	18
Figure 2.3.	How Precision and Rounding Error Affect the Outcome. ....	22
Figure 2.4.	Shadowing Lemma Effect and Numerical Stability. ....	26
Figure 2.5.	Enlarged View of the Shadowing Lemma Effect. ....	27
Figure 3.1.	Non-monotonicity Appears Throughout the Original Dewar Model. ....	34
Figure 5.1.	Standard Deviation of Response Surface Points When $n = 1000$ . ....	41
Figure 5.2.	Standard Deviation of $Y = (\hat{p}_{i+1} - \hat{p}_i)$ ....	44
Figure 5.3.	A Piecewise Linear, Non-monotonic Curve. ....	47
Figure 5.4.	Z-value Transformation Function. ....	50
Figure 5.5.	Adjustment Factor for z Statistic Does Not Affect Results ....	52
Figure 5.6.	A Constant, Monotonic Function. ....	54
Figure 5.7.	A Decreasing, Monotonic Function. ....	55
Figure 5.8.	An Extremely Non-monotonic Function. ....	56
Figure 5.9.	What Happens If the Frequency Is Doubled? ....	57
Figure 5.10.	A Non-monotonic, Piecewise Function. ....	58
Figure 5.11.	Histograms of NmPF Parameter Values For a Surface. ....	61
Figure 5.12.	Known Response Curves Used in Full Factorial Design Experiment. ....	63
Figure 5.13.	Is the Response Curve Non-Monotonic or Not? ....	71
Figure 5.14.	A Close-up View of Figure 5.13. ....	72
Figure 5.15.	$\lambda$ Reduces the Effect of Random Noise on the NmPF Values. ....	73
Figure 5.16.	NmPF Parameter Values and Extreme Non-Monotonic Behavior. ....	74
Figure 5.17.	The Effect of $\lambda$ on NmPF Parameter Values and Extreme Non-monotonic Behavior. ....	75
Figure 6.1.	Pairs Plot: Response Variables From the Fractional Factorial Experiment. ....	80
Figure 6.2.	Explaining Correlation. ....	82
Figure 6.3.	Explaining Negative Correlation Among NmPF Parameter Values. ....	83
Figure 6.4.	Checking the MANOVA Model For Normal Residuals. ....	85
Figure 6.5.	Effects of Stochastic Parameters on Mean Value of NmP. ....	86
Figure 6.6.	Comparing Original Dewar Model With Stochastic Model. ....	90

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF TABLES

Table 0.1.	Latin Hypercube Sampling Results. ....	xxi
Table 2.1.	Original Dewar Model Parameters. ....	13
Table 2.2.	Using Double Precision, Rounding Error Affects the Outcome After Only 3278 Iterations. ....	21
Table 3.1.	Latin Hypercube Sampling Experiment Design and Results.....	32
Table 4.1.	2 <sup>9-3</sup> Resolution V Fractional Factorial Experimental Design. ....	38
Table 5.1.	NmPF Parameter Means, Standard Deviations and Quantiles For the Worst Case Monotonic Curve.....	59
Table 5.2.	NmPF Parameter Means, Standard Deviations and Quantiles For the Worst Case Monotonic Surface. ....	62
Table 5.3.	ANOVA Summary For the NmP Parameter.....	64
Table 5.4.	ANOVA Summary For the J Parameter. ....	65
Table 5.5.	ANOVA Summary For the SJ Parameter. ....	66
Table 5.6.	ANOVA Summary For the NmT Parameter. ....	66
Table 5.7.	ANOVA Summary For the T Parameter. ....	67
Table 5.8.	ANOVA Summary For the ST Parameter. ....	68
Table 6.1.	Response Variable Correlation Matrix. ....	81
Table 6.2.	Summary of MANOVA Linear Model with Interactions.....	84
Table 6.3.	Calculated Effects of the Stochastic Parameters.....	86
Table 6.4.	Optimized Effects of Model Parameters on NmPF Parameters.....	89
Table 6.5.	Mean Values of the NmPF Parameter Values that Result from the SIP Solution.....	90
Table B.1.	NmPF Full Factorial Experiment Results.....	105
Table C.1.	Results of Fractional Factorial Experiment .....	107

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF SYMBOLS, ACRONYMS, AND/OR ABBREVIATIONS

$\lambda$	An argument to the NmPF
AC	Attrition Coefficients
DoD	Department of Defense
FRRT	Force Ratio Reinforcement Threshold
FRWDT	Force Ration Withdrawal Threshold
IF	Initial Force Level
J	Jumps, a measure of non-monotonicity calculated by the NmPF
M&S	Models and Simulations
NmP	Non-monotonic Parameter, a measure of non-monotonicity calculated by the NmPF
NmPF	Non-monotonic Parameter Function, quantifies six measures of non-monotonicity in stochastic response surfaces.
NmT	Non-monotonic Trends, a measure of non-monotonicity, calculated by the NmPF
PIFRT	Percent of Initial Forces Remaining Reinforcement Threshold
PIFWDT	Percent of Initial Forces Remaining Withdrawal Threshold
RBA	Reinforcement Blocks Available, the total number of reserve units
RBD	Reinforcement Block Delay, time delay until reinforcements arrive
RBS	Reinforcement Block Size, number of troops in each reinforcement unit
SJ	Significant Jumps, a measure of non-monotonicity, calculated by the NmPF
SIP	Simple Integer Program
ST	Significant Trends, a measure of non-monotonicity, calculated by the NmPF
T	Trends, a measure of non-monotonicity, calculated by the NmPF
THRESH	An argument to the NmPF

THIS PAGE INTENTIONALLY LEFT BLANK

## ACKNOWLEDGMENTS

An amphibious creature (U.S. Marine) by nature and trade for more than twenty one years, I languished in a slough of mental despondence until ordered by Headquarters U.S. Marine Corps to earn a Master's Degree in Operations Research at the Naval Postgraduate School. I would like to thank all of my professors for their part in the intellectual renaissance I have experienced during the past two and a half years—who says you can't teach an old dog a new trick, or two? I now return to the swamp from which I emerged a short time ago. If you look carefully, as I sink silently beneath the surface, you will note a gleam in my eye, the glint of a promise to return, someday, to the world of academia and to the chase of those elusive beasts, original ideas.

Among all the teeming billions what is the probability that we should come together in this time and place? Surely such a thing is not normal. Since random events colluded to bring us together, it is meet that at least some semblance of a random event should occur at our parting. It has been an honor and a pleasure to be your student.

```
set.seed(12345678)
profs <- sort(profs)
index <- sample(1:length(profs),length(profs),replace=F)
profsList <- profs[index]
```

1	Gordon	Bradley	13	Arnie	Buss
2	Jim	Eagle	14	Carlos	Borges
3	Craig	Rasmussen	15	Rob	Dell
4	Tim	Anderson	16	Gerald	Brown
5	Matt	Boensel	17	Sam	Buttrey
6	Hal	Fredricksen	18	Robert	Koyak
7	Patricia	Jacobs	19	Robert	Read
8	Siriphong	Lawphongpanich	20	Dave	Canright
9	Tom	Lucas	21	Wayne	Hughes
10	Bill	Gragg	22	Paul	Sanchez
11	Beny	Neta	23	Toke	Jayachandran
12	Hal	Blanton			

Special thanks to:

My thesis advisor, Tom Lucas, for allowing me the latitude to do it my way, and yet, more than once, for keeping me from rushing headlong over a precipice I failed to notice as I scurried about in the dark night of ignorance.

My wife for her patience, the mounds of cookies, vats of coffee, and innumerable back rubs, as I spent night and day, for many months, hunched over the computer in the corner of our room, ignoring the rest of the world.

The four inspirations of my life, Neriza, Nelma, Joseph and Isaac who, along with their mother, have given my life warmth and joy in abundance.



THIS PAGE INTENTIONALLY LEFT BLANK

## EXECUTIVE SUMMARY

Decision makers, throughout the Department of Defense (DoD), use models and simulations (M&S) when choosing between various strategic courses of action, whether or not to purchase particular weapons systems, implement programs, terminate existing programs, close existing facilities, open new facilities, and so on. The cost of these weapon systems, programs and facilities can run into the billions of dollars, and there is always the possibility of lives lost. Ensuring that these M&S are logically coherent and provide results that are consistently valid for the purposes for which they are employed is of paramount importance. To the degree that these M&S cannot be verified and validated, decision makers risk making bad decisions. Bad decisions increase the risk of wasting time, money, and resources; they also may put the lives of those affected by these decisions at greater risk.

"For want of a nail...the war was lost," presents a compelling example of chaotic behavior on the battlefield. Had the commander known that victory lay in the possession of a nail, he might have won the war. However, it is an inescapable fact that no matter how careful our measurements, the data used in our analyses are subject to errors. In chaotic systems, even if the magnitude of these errors is extremely small, the uncertainty associated with the errors, conjoined with sensitivity to initial conditions—a characteristic of chaotic systems—creates uncertainty about our knowledge of the system in the future. This uncertainty grows larger as we look farther into the future

No M&S can capture all the intricacies of the real world. However, military analysts seek to capture the essential aspects of combat in their models. The hope is that the output of the models may tell them something useful regarding alternative courses of action, deciding between new weapons systems, and so on. One of the essential aspects of combat may be chaos. In 1991, Dewar et al. showed that a simple, deterministic combat model had a chaotic battle trace. A battle trace is the record of each side's force level through time. Dewar et al. showed that this chaotic battle trace caused non-monotonic behavior in the model. In a combat model, non-monotonicity can be thought of as "capability added to the side of one combatant leading to a less favorable result for

that side.” Repeating Dewar’s note of caution, the chaotic behavior manifested in a combat model may or may not be an accurate reflection of chaos on the battlefield. However, a combat model that exhibits chaotic behavior in an appropriate way, seems, on an intuitive level, to be more realistic than a model that does not.

Figure 0.1<sup>1</sup> shows the non-monotonic output of the original Dewar model. In this two-dimensional graph, initial Red force levels vary from ten to 3500, in increments of ten. Initial Blue force levels vary from ten to 2000, also in increments of ten. The black region in the graph represents those initial force levels that result in a Red win. Now, consider the following two scenarios:

a. In the first scenario, Blue forces are fixed at 900 (this constant Blue force level is indicated by a red line on the graph), and Red forces vary from ten through 3500. As additional troops are added to the Red force, and other variables are held constant, the trend is monotonic. At an initial force level of about 2800 troops, Red starts to win. After that, adding troops to Red’s initial force level results in continued Red wins, just as one would expect.

b. In the second scenario, suppose that, instead of 900, Blue is fixed at 450, while Red varies from 700 to 1800. Then, the response trend goes from Blue wins to Red wins, back and forth many times. Clearly, the trend is non-monotonic. This non-monotonic trend would seem to make it impossible for a decision maker to decide whether or not adding more Red Forces is a good idea.

Note the broad region of extreme non-monotonicity in Figure 0.1. Also note that large portions of this subspace contain very nice monotonic regions. If this graph is any indication of the subspaces that exist in larger models, then it is easy to see that extreme non-monotonicity might go unnoticed, even when it exists. In larger models, the dimensionality of the phase space is incomprehensibly vast. It is entirely possible that these large models are operating in purely monotonic regions. However, it is also possible that they are teetering on the edges of non-monotonic regions like the one pictured here.

---

<sup>1</sup> Most of the graphs and tables in this thesis are best viewed in color. The original version of this document is available in PDF format from the Naval Postgraduate school, <http://www.nps.navy.mil>.

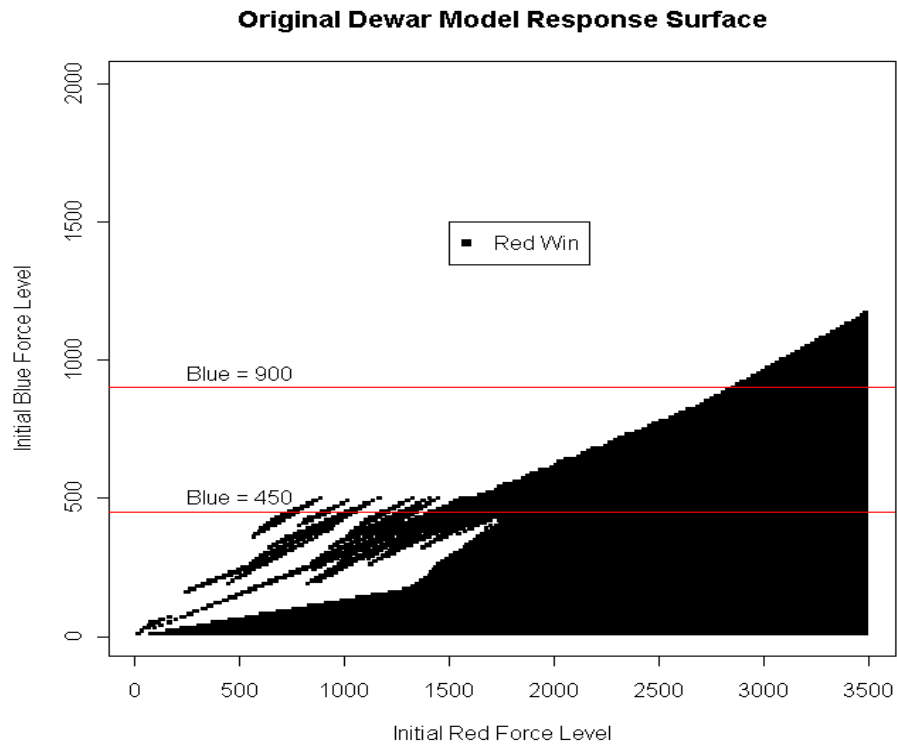


Figure 0.1. The Response Surface of the Original Dewar Model.

When a model, used to decide between alternative courses of action, exhibits non-monotonic behavior, the validity of the model comes into question. For many years, analysts have been plagued by non-monotonicities in their models. The causes of non-monotonicities in the output of M&S are usually traceable to the effects of nonlinear interactions. For example, the interaction between reinforcement and attrition in combat models is nonlinear. In the Dewar model, this nonlinear interaction manifests itself as a chaotic battle trace. The battle trace, in turn, causes non-monotonic trends in the response surface. Although the Dewar model is quite small relative to many of the models currently used by the Department of Defense, it includes many of the same processes present in these much larger models. Thus, it is reasonable to suspect that the larger models may also be plagued by chaos and its consequent non-monotonicity.

Due to the difficulties inherent in the analysis of nonlinear systems and the immense number of parameters in these large models, the traditional reaction has been to ‘fix’ the model until the non-monotonic behavior goes away. These fixes often attenuate

the realism of the model, making the output less useful (Dewar et al.,1991). Another ‘solution’ has been to just ignore the non-monotonicities as anomalous: “It’s just a quirk of the model.” However, the nonlinearities causing the non-monotonicity may be an essential part of the model. In this case, the non-monotonicities should not be ignored or ‘fixed’ so that they go away. Advances in computing power and developments in chaos theory over the last thirty years have made it possible to look at the problem of non-monotonic behavior from a different perspective. Perhaps there is a way to retain the inherent nonlinear nature of the thing being modeled and still obtain an interpretable response surface that can be used to help decision makers decide between competing alternatives.

The dozen previously published papers on the Dewar model examined only small subsets of its 18 dimensions. The combinatorial possibilities of main effects and interactions among the 18 dimensions are too great to examine en masse. There are some who would caution against spending time on pragmatically useless models. However, intensive study of the Dewar model has the potential to uncover many useful insights into processes present in much larger and more complex models. Consequently, we have three goals. First, look broadly across the Dewar model to determine if non-monotonic behavior is wide-spread or only an anomaly in the few dimensions previously studied. Second, determine if the non-monotonicity can be abated through stochastic perturbation. Third, in order to accomplish the second objective, develop a method for measuring non-monotonicity in the response surfaces generated by the model.

Our overall goal is to discover if the trend of the response surface in non-monotonic regions can be made more amenable to interpretation, without destroying the chaos inherent in the attrition/reinforcement process. While this might have the appearance of just ‘fixing’ the model, there is a qualitative difference between artificially scripting a model’s processes—scripting removes, to a great extent, any semblance of reality—and allowing a model’s parameters to reflect the randomness inherent in combat. An example of this randomness is the uncertainty a commander faces when trying to decide when to send in his reserves or, if discretion is the better part of valor, when to withdraw.

Latin Hypercube Sampling (McKay 1979), an extension of Latin Square sampling, is used to thoroughly search the 18 dimensions of the Dewar model. This search discovers non-monotonicity in not only five additional two dimensional subspaces, but also in multiple measures of effectiveness (MOE's). The second column in Table 0.1 displays the results of the Latin Hypercube search through each of the natural two-dimensional subspaces. The first number in the second column corresponds to the initial forces subspace, where the response surfaces, generated by each of the sixteen design points, all exhibited non-monotonic trends, in some cases more extreme than in the original Dewar model. The numbers, in the second column, indicate the percentage, from each design point, that were non-monotonic response surfaces. This table indicates that non-monotonicity is pervasive in at least six of the two-dimensional subspaces of the deterministic Dewar model.

<b>Two-dimensional Dewar Model Subspaces</b>	<b>Percent of Response Surfaces Containing Non-monotonic Behavior</b>
Initial Force Levels	80%
Force Ratio Reinforcement Thresholds	81%
Percent of Remaining Forces Reinforcement Threshold	18%
Force Ratio Withdrawal Threshold	0%
Percent of Remaining Forces Withdrawal Threshold	0%
Reinforcement Blocks Available	6%
Reinforcement Block Delay	100%
Reinforcement Block Size	88%
Attrition Coefficients	94%

Table 0.1. Latin Hypercube Sampling Results.

78 of the 144 response surfaces found during the Latin Hypercube Sampling exhibited non-monotonic behavior.

Striking examples of the wide-spread non-monotonicity in the Dewar model can be seen in Figure 0.1 (the original Dewar model response surface), and Figure 0.2. Figure 0.1 exhibits non-monotonicity with respect to the measure of effectiveness (MOE), 'who wins.' Two of the many striking examples we discovered, using Latin Hypercube Sampling, are displayed in the top row of Figure 0.2. These two graphs exhibit non-monotonicity with respect to a second MOE, 'length of battle.' The two bottom graphs in Figure 0.2 exhibit non-monotonicity with respect to the MOE 'who wins.' All of these graphs represent projections of the 18 dimensions of the Dewar

model onto the two-dimensional subspaces indicated by the titles of each graph. Each of these response surfaces is generated by the deterministic Dewar model.

## *Two-Dimensional, Non-monotonic Subspaces*

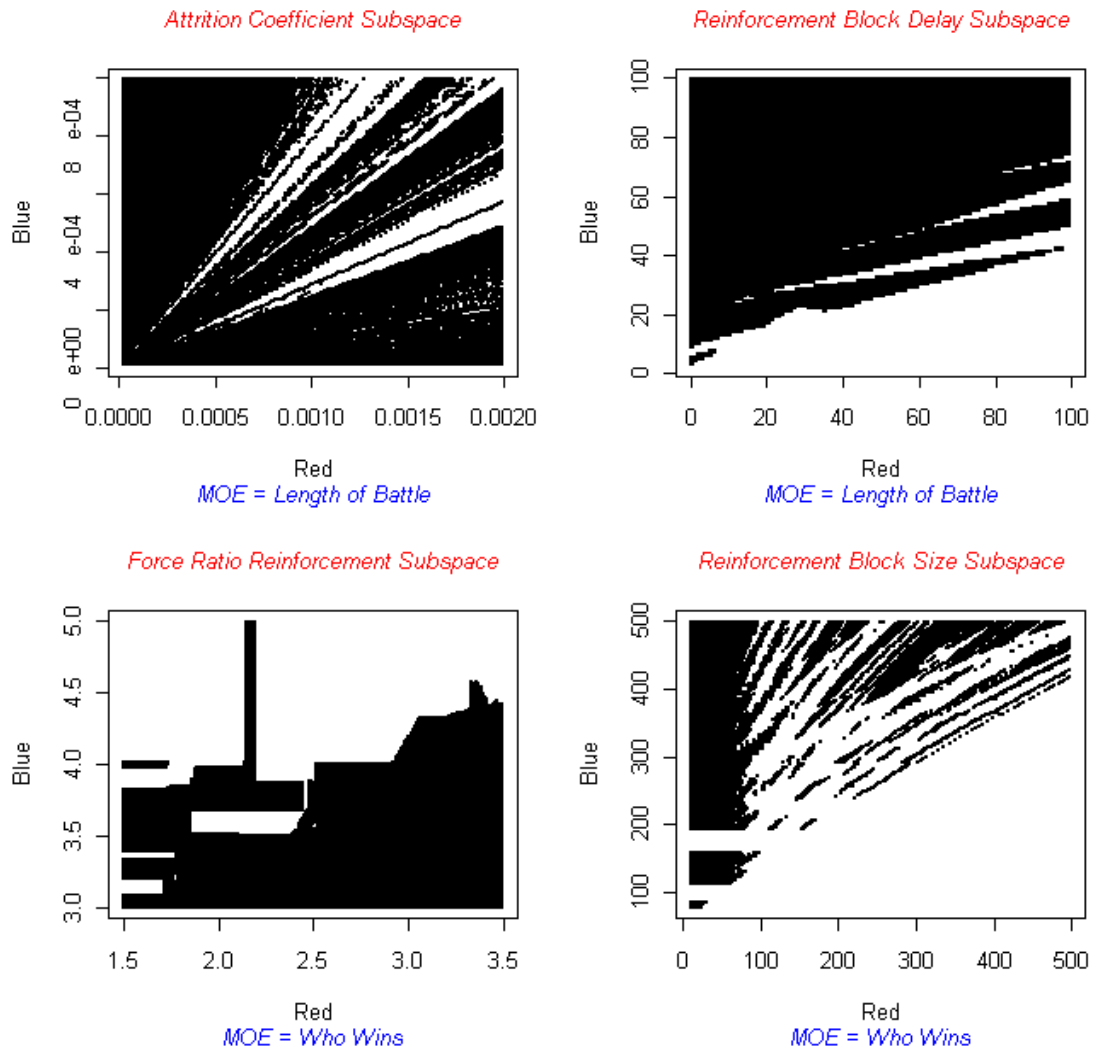


Figure 0.2. Two Dimensional Subspaces Exhibiting Non-monotonic Behavior.

These four graphs are representative of the non-monotonicity found across broad regions of the Dewar model's hyperspace.

The Latin Hypercube Sampling was conducted to determine if there were any additional non-monotonic subspaces of the Dewar model. The fact that there are many more such subspaces is cause for concern in the modeling and simulation community. The Dewar model contains the same basic processes that many of the larger models use,

such as decision thresholds and attrition processes. If the interaction of these processes in the Dewar model generates such widespread non-monotonic behavior, then the larger, more complex models may also be affected by similar non-monotonicities, perhaps caused by some underlying, as yet, undiscovered chaos. It has been shown that recent output of at least one of these models has exhibited non-monotonic behavior (Saeger, 2000).

The fact that there is non-monotonicity throughout the Dewar model gives added impetus for finding a way to deal with it. What can be done to smooth non-monotonicities in the response surface caused by the chaos in the battle trace? Much of the literature reviewed for this paper indicates that stochastic modeling can be a useful way to deal with non-monotonic behavior of not only the simple combat model studied by Dewar et al., but also other, more complex models.

A fractional factorial design experiment is run to determine the effect of stochastic modeling on the trend of the response surface of the Dewar model. The results are dramatic and convincing and support previous work done by Allen (1994), Lucas (1997) and Seager (2000). Stochastic perturbation can both reduce and exacerbate the non-monotonic behavior of the response surface. The model parameters having the greatest effect on the reduction of the non-monotonic behavior in the model are the attrition coefficients. When the attrition coefficient parameters are made stochastic, the average reduction in non-monotonicity is four orders of magnitude. It is also important to note that some of the response surfaces, generated during the fractional factorial experiment, resulted in an increase in non-monotonicity of two orders of magnitude. Figure 0.3 displays the best, and worst, response surfaces generated by the fractional factorial experiment. The graph on the left is an example of poor modeling decisions exacerbating the non-monotonicity in the response surface. The graph on the right is an example of the dramatic reduction in non-monotonicity and the improvement in the interpretability of the response surface when stochastic modeling is done carefully. This second graph also appeals to our intuition—the outcome remains uncertain until one side or the other quits the field of battle.



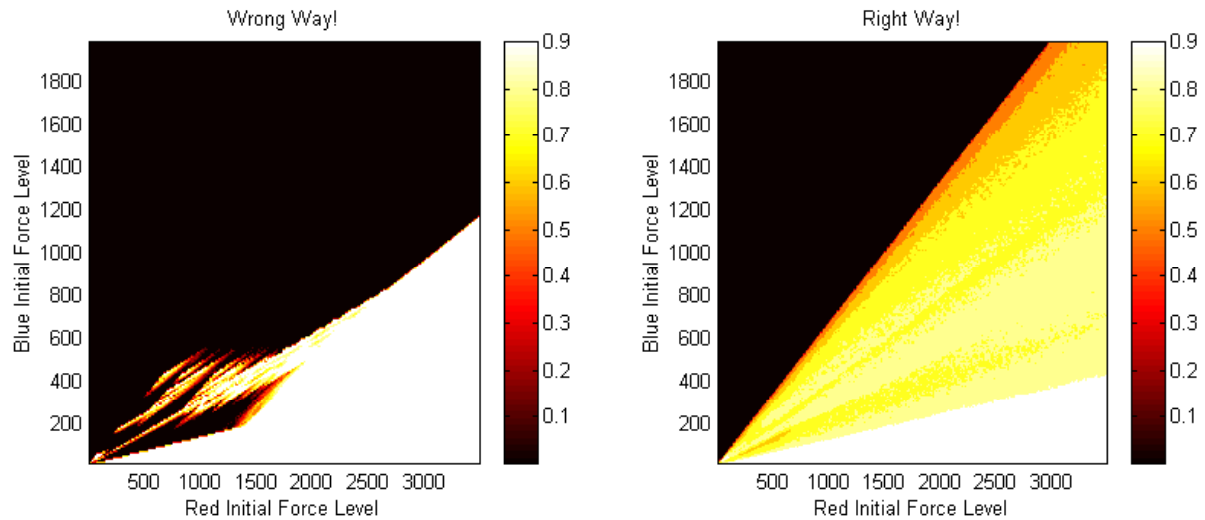


Figure 0.3. Careful Stochastic Modeling Can Smooth Non-monotonic Trends. The graph on the left is an example of the effect of, haphazardly making parameters stochastic. This graph exhibits even more non-monotonicity than the original Dewar model response surface. At the opposite extreme, when done the right way, as in the graph on the right, the non-monotonic behavior of the response surface can be dramatically reduced, making it more amenable to interpretation.

*Caveat actor et cavendo tutus:* The Dewar model is a relatively small model when compared to most of the larger, more complex models that DoD currently uses. Counting explorations, testing, Latin Hypercube Sampling, fractional factorial experiments, and final empirical testing of analytical results, approximately *ten billion* model runs were made during the course of this study. This is but an inconsequential fraction of the number of runs it would take to thoroughly explore every corner of the 18-dimensional Dewar model. Clearly, conducting this type of analysis with the large DoD models is impossible. However, the results of this study may suggest useful methods for dealing with non-monotonic output in these larger, more complex models.

The bottom line is that non-monotonicity may be more pervasive in combat models than previously suspected. Stochastic modeling can be a viable method for dealing with non-monotonic response surfaces. However, stochastic modeling must be done carefully. When it is, the non-monotonic behavior of the model can be dramatically reduced, thereby making the trend of the response surface more useful for comparative analyses. Thus, the realism of the model can be retained while, at the same time, the

trends of the response surface can be used to support decisions between competing alternatives.

Feigenbaum, Mandelbrot and other pioneers of chaos theory have shown that chaos is the rule rather than the exception in the real world (Gleick, 1987). The possibility that chaos and its consequent non-monotonicity are also the rule rather than the exception in large, complex combat models, is very real. Stochastic modeling shows some promise as a way to deal with this non-monotonicity without destroying the chaos that, intuition tells us, adds to the model's realism. More realistic models can help decision makers save time, money, other assets—and, perhaps, another soldier, sailor, airman or Marine gets to enjoy his pension.

THIS PAGE INTENTIONALLY LEFT BLANK

# **I. INTRODUCTION**

## **A. PREAMBLE**

Decision makers throughout the Department of Defense (DoD) use models when choosing between various strategic courses of action and deciding whether or not to purchase particular weapons systems, implement programs, terminate existing programs, close existing facilities, open new facilities, and so on. The cost of these weapon systems, programs and facilities can run into the billions of dollars, and there is always the possibility of lives lost. Ensuring that these models and simulations are logically coherent and consistently valid for the purposes for which they are employed is of paramount importance. To the degree that these models and simulations are not verified and validated, decision makers may make bad decisions, which wastes time, money, and resources and possibly risks the lives of those affected by these decisions.

In 1991, RAND released a study indicating that even simple, deterministic combat models are subject to non-monotonic behavior<sup>2</sup>. This finding was not new. However, the finding that the non-monotonicity was due to chaos inherent in the model was new. This is cause for concern in the modeling and simulation community as more complex models also may be affected by chaotic behavior. Many models used in the Department of Defense are highly complex and nonlinear, with thousands of variables. It has been shown that recent output of at least one of these models has exhibited non-monotonic behavior (Saeger 2000).

In 1931, Kurt Godel proved the possibility that a sufficiently complex system can exhibit synergistic behavior not explainable by the sum of the system's components (Nagel, 1958). Looking at the problem of complexity from a slightly different angle, Ancker (1995) states that “the most successful procedure for model building [in engineering and the physical sciences] has been to work from the simple and small to the larger and more complex. In combat modeling, the process has usually been to start large.” He suggests “that this trend should be reversed.” The implication here is that

---

<sup>2</sup> Capability added to the side of one combatant leads to a less favorable result for that side.

analysts need to thoroughly understand the smaller, simpler models before proceeding to the larger, more complex models. The RAND study was a step in the right direction. This thesis seeks to take the RAND study a step further. It examines the Dewar model in greater detail, with the goal of using insights gained through greater understanding to help make models and simulations more useful to decision makers. Better models and greater understanding, in turn, may help save time, money, resources and lives.

## **B. CHAOS AND NON-MONOTONICITY IN COMBAT MODELS**

"For want of a nail . . . the war was lost" provides a compelling example of chaotic behavior on the battlefield. Had the commander known that victory lay in the possession of a nail, he might have won the war. However, it is an inescapable fact that no matter how careful our observations, the data used in our analyses are subject to errors in measurement. In chaotic systems, even if the magnitude of these errors is extremely small, the uncertainty associated with the errors, conjoined with sensitivity to initial conditions—a characteristic of chaotic systems—creates uncertainty about our knowledge of the system in the future. This uncertainty grows larger as we look farther into the future (see Figure 1)<sup>3</sup>. Many diverse disciplines use computer modeling and simulation to answer questions such as: "What are the possible future states of this system given these initial conditions?" Hence, the causes and effects of chaos on the monotonic behavior of computer simulations and models should be of great interest to those involved in such research.

Many of the models currently used by the Department of Defense are coded, populated with data, and run time after time in order to predict future states of various systems. But, lurking beneath the surface of these models is the unpredictable, dynamic, unstable realm of chaos, resisting analysts' efforts to obtain reliable results. The instability inherent in nonlinear dynamical systems has been known for hundreds of years. However, only since the 1970s has chaos been studied in any detail (Gleick, 1987). Now, thirty years later, renewed interest in the "New Sciences" is stirring in the world of Military Operations Research, (Palmore, 2000). One cause of this renewed

---

<sup>3</sup> Most of the graphs and tables in this thesis are best viewed in color. The original version of this document is available in PDF format from the Naval Postgraduate school, <http://www.nps.navy.mil>.

interest is the dramatic increase in computing power available to the operations research analyst in the last ten years.

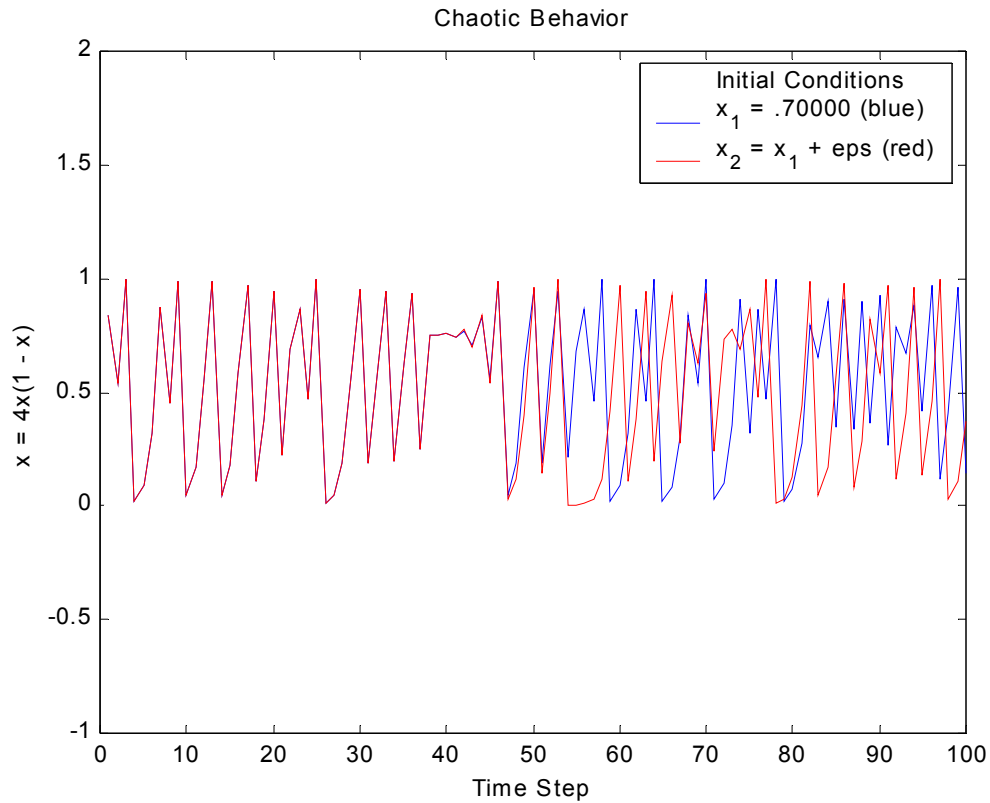


Figure 1.1. A Trait of Chaos: Sensitivity to Initial Conditions.

The value  $eps \cong 2.2^{-16}$ . Yet, even when this simple system's initial value changes by such a small amount, the future state cannot be predicted indefinitely into the future. In this very simple recursive equation, one of the attributes of chaos, sensitivity to initial conditions, is manifest somewhere between the 40<sup>th</sup> and 50<sup>th</sup> time step. The values at each time step are calculated by the Logistic Iterator which is known to be chaotic. The y-axis of the graph is labeled with the recursion formula.

As computers have become more powerful, many DoD models and simulations have become more complex. Increased complexity requires analysts to be much more cautious with regard to the output of their models and simulations. At the same time, they are able to use the increased power of computers to more fully explore the concepts of chaos and complexity. So, while the evolution of the computer has exposed analysts more directly to the effects of chaos, it has also given them a means with which to study and explore its attributes.

In the realm where models and simulations (M&S) are used to support decision making, the decision maker must be able to rely on the results of M&S. Verification, validation and accreditation (VV&A) is a process whereby M&S are sanctioned for a particular use. Verification ensures that the models are properly coded. Validation ensures that the models are appropriate for a particular purpose in a “real-world” setting. Accreditation is the official sanction for use by the decision maker who will use results from the M&S in his decision making process. VV&A is the process whereby the decision maker ensures that his decisions, based on M&S results, are rationally defensible.

The output of a deterministic simulation or model is called a response curve when it is plotted against a single input parameter. If more than one parameter is varied then the plotted output is called a response surface. Monotonic behavior means that there are no bumps, dips, or unexpected reversals in the response surface. Mathematically, monotonic behavior is defined by the following: for discrete functions, if a function  $f$  is monotonic non-decreasing, then, if  $x_1 \leq x_2$ , then  $f(x_1) \leq f(x_2)$ . Similarly, if  $f$  is monotonic non-increasing, then, if  $x_1 \leq x_2$ , then  $f(x_1) \geq f(x_2)$  for all  $x$  in the domain of  $f$ . Another way of defining this is to say that if  $f$  is a monotonic non-decreasing function, then first differences are greater than or equal to zero, i.e.,  $f(x_2) - f(x_1) \geq 0$ . Continuous, monotonic non-decreasing functions have first derivatives that are nonnegative throughout their domains. Knowing whether or not a model’s output is

monotonic can be vitally important, as decision makers use this output to compare competing strategies, tactics, systems, and so on.

### **C. PREVIOUS RESEARCH PERTINENT TO THIS STUDY**

In 1991 Dewar, Gillogly and Juncosa published a paper that caused a significant response in the combat modeling and simulation world. A stream of responses, papers and studies followed in its wake. Each one argued for or against the main points made by Dewar et al. or raised interesting issues closely related to the main theme of the paper. It is a well-known fact that chaos is present in combat models. Chaos can and often does cause non-monotonicity in the output of combat models. The concern with this is that

. . . when comparisons of strategy, tactics or systems are based on a combat model that depends on monotonic behavior in its outcomes, modeling combat decisions based on the state of the battle must be done very carefully. Such modeled decisions can lead to non-monotonic and chaotic behavior and the only sure ways (to date) of dealing with that behavior are either to remove the modeled decisions or to verify that the model is monotonic in the region of interest (Dewar et al., 1991)

Clearly, when decisions depend on a model's monotonic behavior, non-monotonicities in the model can make comparisons difficult, if not impossible.

Over the next six years, there were no fewer than ten articles and letters to the editor in *Phalanx* (Bulletin of Military Operations Research), not to mention several longer, more in-depth studies conducted by Julian Palmore at the University of Illinois and Thomas Lucas of RAND, among others. Each of these papers explored ways to explain, overcome or attenuate the effects of chaos and the resulting non-monotonicities.

Tsai and Ellenbogen (1992) more fully explored the idea of finding tighter bounds on non-monotonic regions in the phase space of models. They simplified, even further, the model used by Dewar et al., in order to come up with a general methodology involving "transition points," which lead to bounds for non-monotonic regions. The authors noted that defining bounds on more realistic models would be "considerably more complex."

Palmore offered three causes of instabilities (non-monotonicities):



1. Computer arithmetic,
2. Structurally unstable implementations of branchings on thresholds set by decision tables, and
3. Structural defects in decision logic (1992).

Palmore discussed several ways to investigate and address these problems.

Louer (1993), in a letter to the editor of Phalanx, recalled non-monotonic mischief in 1971, when he was involved in the development of the Concepts Evaluation Model (CEM) for the U.S. Army. He advocates the solution they came up with—parametric variation of variables and stochastic decision thresholds. Louer claims that both of these methods cope with the underlying chaos, while providing results that are very informative and trends that are monotonic in behavior.

Allen, Gillogly and Dewar of RAND (1993) disagreed with Louer's comments and provided a counterexample to “demonstrate that stochastic thresholds not only can fail to eliminate non-monotonic behavior, but can actually make the non-monotonic behavior worse.” Louer's response was that Allen et al. had missed the point of his letter, and he went on to emphasize the use of experimental design

. . . to develop response curves reflecting the force performance over these ranges of uncertainty . . . If non-monotonic effects show up in these response curves, they can be identified. The skilled analyst then needs to make an interpretation of these effects, locate where the effects come from and assess if they have any significant influence on the trends of the response function curves (1994).

Cooper picked up the gauntlet cast by Allen et al. and defended Louer. He pointed out that Allen's experiment was flawed in its design, that the stochastic thresholds Allen included in his model were “masked” by other model constraints. Thus, “within an unreachable domain, whether a value in that domain is chosen deterministically or stochastically does not matter.” Cooper went on to emphasize what Louer originally urged “ . . . far-ranging sensitivity trials, to explore more of a model's domains . . . for ever more cases and more analysis until there is understanding” (June, 1994).

In a follow-on article, Cooper (September 1994) noted that the decision rules in the Dewar model could be improved. He showed how a simple, minor change to the Dewar model results in purely monotonic behavior. If the decision to withdraw is made with respect to not only engaged forces but also reserve forces, then the non-monotonicities disappear.

Huber and Tolk (1994) responded to Allen et al.'s response to Louer's comments on stochastic decision thresholds. The counter-counterexample showed that "non-monotonic effects may largely be eliminated if dynamic *mission-oriented* decision rules are used rather than the static *state-oriented* decision thresholds considered by Dewar, et al." Their conclusions echoed Cooper's with regard to the unrealistic nature of the decision thresholds of the Dewar model and also noted the problems with Allen's "masked" decision threshold.

Lucas (1997) conducted a study which demonstrated that how you make the decision thresholds stochastic matters. He studied not only stochastic decision thresholds, but also stochastic attrition coefficients. Lucas concluded that careful stochastic manipulation of both decision thresholds and attrition coefficients can significantly smooth the non-monotonicities that arise due to dynamic instabilities inherent in combat models.

Johnson, Isensee and Allison (1995, pp. 87-100) conducted a stochastic experiment on the Army's Concept Evaluation Model (CEM) that indicated "which stochastic features most influenced the variability among replications of one simulated campaign and [outlined] costs and benefits of using a stochastic version of the CEM."

The last article in Phalanx (Lucas and Allen, 1997) highlighted the differences between the two camps of analysts that had formed on either side of this ongoing discussion about non-monotonic behavior of combat models. Lucas and Allen examined whether or not non-monotonicities can be tamed by making the decision thresholds in the model stochastic vice deterministic. This joint paper concluded that

the problem of non-monotonicity in even simple deterministic or stochastic models continues to be a problem that needs to be addressed.

Investigation of the types of randomization that usually provide sufficient overlap to produce ‘smoother’ results may be a useful line of inquiry.

Even more recently, the Military Operations Research Society published a special edition of its Journal dedicated solely to the question of "Warfare Analysis and Complexity - The New Sciences, MORS (2000)." In this special edition, Lucas (2000) outlined the divide between deterministic and stochastic models and gave several examples where stochastic modeling was superior to deterministic modeling. Lauren (2000) examined the "possibility of using a fractal to describe attrition" in a combat model. Using historical data and simulation, he showed that attrition can reasonably be modeled in such a manner. Hausken and Moxnes (2000) derived a set of ordinary differential equations using the theory of stochastic difference equations. They applied their stochastic model to the Ardennes Campaign of 1944 to provide "a more descriptive presentation of the campaign."

At a recent MORS Symposium, Saeger et al., (2000) gave a briefing on recently discovered non-monotonic behavior in the U.S. Army's VIC (Vector-in-Command) model. After a closer look, they determined that the model was properly coded and the data were correct. In an attempt to get meaningful results out of the model, they defined a neighborhood of the phase space, randomly perturbed selected variables and attempted to fit a probability curve to the output values of multiple runs. They concluded that "the probability distribution of the [response curve] is a function of the variable that is perturbed. [But there is] no obvious way to determine, *a priori*, which variables to perturb [nor] by what magnitude to perturb them." Their recommendation was to "perturb all variables [and] perform sensitivity analysis in perturbation magnitude." In a high-dimensional model, such a sensitivity analysis would be challenging, to say the least.

Much of the literature reviewed for this paper indicates that stochastic modeling may provide a useful way to deal with non-monotonic behavior of not only the simple combat model studied by Dewar, et al., but also other, more complex models. Clearly, defining the link between model structure and its output, in the presence of chaos, is a

small, but very essential part of model verification. Once we have obtained a more comprehensive understanding of simple models, then we can proceed to investigate, step by step, the more complex models.

## **D. METHOD**

This study is broken up into several smaller pieces, each of which is briefly summarized below.

### **1. Java, Numerics and the Effect of Finite Precision on the Battle Trace**

Since we are trying to ascertain the effect of stochastic perturbation on the response surface of the Dewar model, the original Dewar model must be modified. Stochastic perturbation is nothing more than drawing random values from a given distribution and assigning these random values to the parameters of the model. Of course, the parameters of these distributions are carefully chosen so that the distributions are centered around the nominal values used in the original model. Modifying the original model in this manner causes random perturbations in the parameter space of the model, which in turn, affects the model's output. The modified stochastic model is re-coded in Java. Following the advice of Palmore (1996), the numerics of the model are examined for the following:

- a. Computer arithmetic,
- b. Structurally unstable implementations of branchings on thresholds set by decision tables, and
- c. Structural defects in decision logic.

### **2. Latin Hypercube Sampling**

Dewar et al. discovered significant non-monotonicity in the response surface of their model. However, they examined only a two-dimensional subspace of their 18 dimensional model. Are there other non-monotonic subspaces in the Dewar model? The implication here is that if the Dewar model exhibits significant non-monotonicity in many of its subspaces, then other, more-complex models may also be hiding such non-monotonic behavior. Additionally, this implies that non-monotonic behavior may be more widespread than previously suspected. Latin Hypercube sampling is used to

systematically explore all 18 dimensions of the Dewar model for non-monotonic behavior.

### 3. Fractional Factorial Experiment

Clearly, trying to examine all possible combinations of even the simple Dewar model, for the effect of stochastic parameter modeling, is infeasible. A full factorial  $2^{18}$  experiment would require  $2^{18} * 69451 * 1000 \cong 18$  trillion model runs to explore all 18 dimensions, at only two levels, without any replications<sup>4</sup>. This would take approximately 249 years on a Pentium III 600Mhz computer with 256 Mb of RAM. The number gets even larger as more levels are added. To keep the number of runs required down to a reasonable number and to extract the maximum amount of information out of each run, a fractional factorial experiment is used to examine the main and first-order interaction effects of stochastic parameters on the Dewar model. This experiment is confined to the same two-dimensional subspace that Dewar et al. examined. However, the Java Code is easily modified to examine any of the other,  $\binom{18}{2} - 1 = 152$ , two-dimensional subspaces of the model. The results of this experiment show that, when done properly, stochastic modeling can significantly smooth the non-monotonicities, making the response surface useful for comparative purposes.

### 4. Defining Non-monotonic Parameters

To analyze the response surfaces generated by the fractional factorial experiment, we need a parameter that measures the non-monotonicity of the response surface. We define, develop, numerically test, and then employ a non-monotonic parameter function (NmPF) which calculates six parameters that measure various attributes of the non-monotonicity of a response surface.

### 5. Measures of Effectiveness

Just as in the Dewar study, “who wins” is the measure of effectiveness used in this study. However, using stochastic parameters changes the response from a binary,

---

<sup>4</sup> The original Dewar model two-dimensional response surface was comprised of 69,451 points. We replicate each of these points 1000 times to generate a stochastic response surface. See Chapter II for details.

win or lose surface, to a continuous response surface representing the probability of a win or a loss. Due to the structure of the model, it is also possible to use other measures of effectiveness, such as ‘length of battle.’ The Latin Hypercube sampling discovers several subspaces of the original Dewar model where the response surface is non-monotonic with regard to this second measure of effectiveness. We do not explore this measure of effectiveness further, but note that it is an area for further research.

THIS PAGE INTENTIONALLY LEFT BLANK

## II. RECODING THE MODEL IN JAVA

What follow are a brief description of the original Dewar model, some comments regarding the modified stochastic version coded in Java, a description of the phase space, or response surface generated by the model, and finally a brief check of the model's numerical stability.

### A. A BRIEF DESCRIPTION OF THE ORIGINAL DEWAR MODEL

The Dewar model contains 18 variable parameters. Therefore the model's output or response is a point in  $\mathfrak{R}^{18}$ . Table 2.1 displays the structure and nominal values used in the original Dewar model. Note the symmetry in the parameters, for each Blue parameter there is a corresponding Red parameter.

Parameter (Variable Names)	Blue	Red
Initial force level (IF)	$B_0$	$R_0$
Reinforcement threshold (FRRT, PIFRT)	$R_n/B_n \geq 4$ or $B_n < 0.8B_0$	$R_n/B_n \leq 2.5$ or $R_n < 0.8R_0$
Maximum allowable reinforcement blocks (RBA)	5	5
Reinforcement block size (RBS)	300	300
Reinforcement delay (RBD)	70	70
Withdrawal threshold (FRWDT, PIFWDT)	$R_n/B_n \geq 10$ or $B_n < 0.7B_0$	$R_n/B_n \leq 1.5$ or $R_n < 0.7R_0$
Combat attrition calculation (AC)	$B_{n+1} = B_n - R_n/2048$	$R_{n+1} = R_n - B_n/512$

Table 2.1. Original Dewar Model Parameters.

The first parameter, initial force level (IF), is the number of troops with which each side starts the battle. As in the original Dewar study, Blue and Red initial force levels typically vary over the range ten through 2000 and ten through 3500 respectively.

There are two reinforcement thresholds for each side: a force ratio threshold (FRRT) and a 'percent of initial force level' threshold (PIFRT). Note that the force ratio for these calculations is the same for both sides—i.e.,  $R_n/B_n$  for both rather than  $B_n/R_n$  for Blue and  $R_n/B_n$  for Red. For the Blue side, if the force ratio goes above four, then Blue calls for reinforcements. Additionally, if the number of remaining Blue forces ever falls below 80% of the initial force level, then Blue calls for reinforcements. Likewise for



Red, if the force ratio ever falls below 2.5 or if the number of remaining Red forces falls below 80% of the initial force level, then Red calls for reinforcements.

Reinforcements come in block sizes (RBS)—that is, the number of reinforcing troops that respond to a call for reinforcements. There is a limit to the total number of reinforcement blocks available (RBA) and, as is usual, there is some delay between the time reinforcements are called for and the time they actually arrive (RBD).

Finally, attrition is determined by the Lanchester Square Law, coupled difference equations in the last row of Table 2.1. The number of remaining Blue troops at the end of the first time step is equal to the number of Blue troops at the previous time step minus the number of Red troops times Blue's attrition coefficient. The attrition coefficient (AC) is the rate at which one Red troop kills Blue troops. The number of remaining Red troops at the end of the  $n^{\text{th}}$  time step is calculated in the same manner.

The structure of the Dewar model represents a large, less effective Red force opposing a smaller, more effective Blue force. Another perspective is that Blue represents an entrenched force defending against a Red attack. In this case, the attacking force usually has a larger force and, due to exposure, is more prone to attrition. The entrenched defending force usually has a smaller force level and a well prepared defensive fire plan; it usually takes fewer casualties and is more efficient at killing, at least initially. Thus, the structure of the Dewar model represents a basic tension between the opposing forces. Care must be taken, when randomly perturbing the model's parameters, to ensure that this basic tension is preserved; otherwise, the model becomes a meaningless function that processes random input and returns meaningless output.

## **B. A STOCHASTIC MODIFICATION AND THE POTENTIAL FOR ADDITIONAL PARAMETERS**

To examine the effect of stochastic modeling on the response of the Dewar Model, the model is re-coded in Java. Each of the parameters discussed above is modeled as a generic random variate so that most distributions, along with their parameters, can be used. This creates the potential for a dramatic increase in the dimension of the model. Each parameter can now be stochastically modeled in  $n$

different ways. The  $n$  distributions have  $m$  parameters that can be considered factors as well. To preclude this explosion of factors, only two distributions are considered. The attrition coefficients are modeled as normal random variates, with means equal to the nominal Dewar model values and standard deviations equal to ten percent of the square root of their means. All other parameters, when stochastic, are modeled as uniform random variates with intervals, centered on the nominal Dewar model values, ranging from plus to minus five percent of the nominal value. These distributions and values are based on the work done by Lucas (1997) and Seager (2000). Clearly, the use of other distributions is possible. Exploring the effects of different distributions on the response surface is another useful area of inquiry, but it is not pursued in this thesis.

### **C. THE RESPONSE SURFACE**

#### **1. A Description of the Deterministic Response Surface**

The original version of the model is deterministic. The values for each of the 18 parameters are passed to the model, which returns a ternary response: “0” if Blue wins, a “1” if Red wins or a “-1” if neither side wins—i.e., the battle has not finished after the allotted number of time steps. To generate a response surface, two of the 18 parameters are allowed to vary over a wide range of values. The points that lie on the response surface are the result of stepping through each of the possible combinations of the two Blue and Red parameters that are allowed to vary. The other sixteen initial parameter values remain fixed for that run. The points, thus generated, are plotted over the first quadrant of the Cartesian plane (all initial parameter values are always positive). The x-axis represents the initial values of the Blue parameter that was allowed to vary. The initial values of the Red parameter that was allowed to vary lie along the y-axis. The z-axis represents the height of each point on the response surface above or below the x-y plane.

Figure 2.1 displays the response surface, of the original Dewar model’s initial force level subspace. This response surface is comprised of 69,451 points. The initial force levels vary from ten through 3500 (Red) and ten through 2000 (Blue), in increments of ten; thus,  $349 \times 199 = 69,451$  points. Obviously, the model must be run 69,451 times to

generate one such response surface. These numbers vary, depending on which parameters are allowed to vary.

Note the region of extreme non-monotonic behavior in Figure 2.1, which occurs when Red initial force levels are between ten and 1800, and Blue initial force levels are between ten and 500. Also displayed in this graph are the force ratio thresholds for reinforcement and withdrawal. Notice that the withdrawal thresholds bound the region of non-monotonicity. Now, we take a look at the response surface generated by the stochastic version of the model.

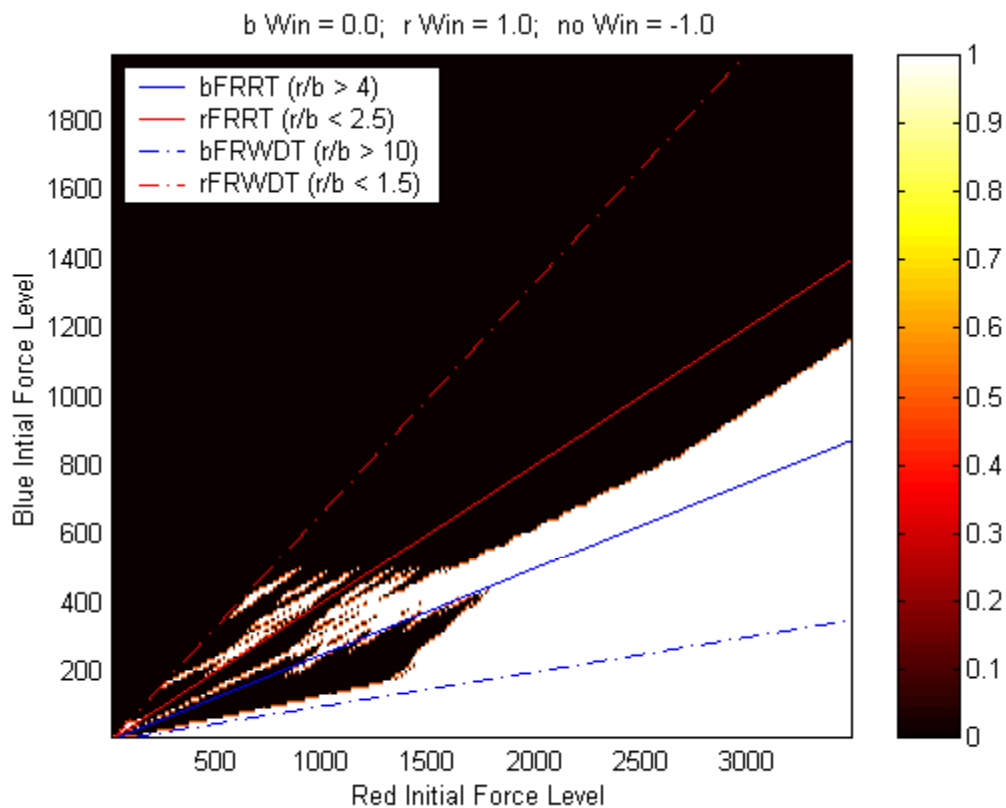


Figure 2.1. Binary Response Surface of Original Dewar Model.

FRRT = Force ratio reinforcement threshold; FRWDT = Force ratio withdrawal threshold. The characters 'r' and 'b' stand for 'Red' and 'Blue,' respectively. This figure displays the response surface of the original Dewar model. The reinforcement and withdrawal thresholds are superimposed on the response surface. Note that the non-monotonicity is bounded by the withdrawal thresholds. The color bar, to the right of the graph, indicates that the black region represents the initial force levels that result in a Blue win. The white region represents those initial force levels that result in a Red win.

## **2. A Description of the Stochastic Response Surface**

The following process generates a stochastic response surface:

a. All stochastic parameters, for a given run, are assigned a random value from the distribution used to model the parameter—attrition coefficients are modeled as normal random variates with means equal to the nominal values of the original Dewar model and standard deviations equal to ten percent of the square root of the mean. All other parameters are modeled as uniform random variates, are centered on the nominal values of the original Dewar model and vary minus to plus five percent of the nominal value.

b. All parameters that are not stochastic are assigned the nominal values of the original Dewar model.

c. Each input point (a vector of the 18 parameter values) is replicated 1000 times. Before each replication is run, a new value is randomly chosen for each stochastic parameter. The reason for the large number of replications is discussed in Chapter V. The short answer, for the moment, is that this reduces the variance of the model's response surface that is due only to the randomness of the stochastic parameters.

The model returns the average of the one thousand replications. Thus, each response point is the mean of a thousand replications of its corresponding input point (an 18-tuple). In the same manner as described above, a response surface is generated by allowing two parameters to vary over a given range. To generate a stochastic response surface, the model must be run  $69,451 \times 1000 = 69,451,000$  times. Since each point on the response surface, for the stochastic version of the model, is the mean of 1000 replications, the response surface is now continuous and represents the estimated probability that Red wins, given the Blue and Red parameter values. Figure 2.2 is an intensity plot of a response surface generated by the stochastic version of the Dewar model. The color bar to the right of the graph gives an indication of the height of the response surface. Note that, in this particular graph, the response surface trend is easier

to interpret than the response surface in Figure 2.1. The region in black represents the initial force levels that result in a Blue win; the region in white represents the initial force levels where Red wins. The colors in between indicate, according to the color scale to the right of the graph, an increasing probability of a Red win. Once again, the force ratio thresholds are superimposed on the graph. The withdrawal thresholds bound the region where the probability of a Red win is greater than zero and less than one.

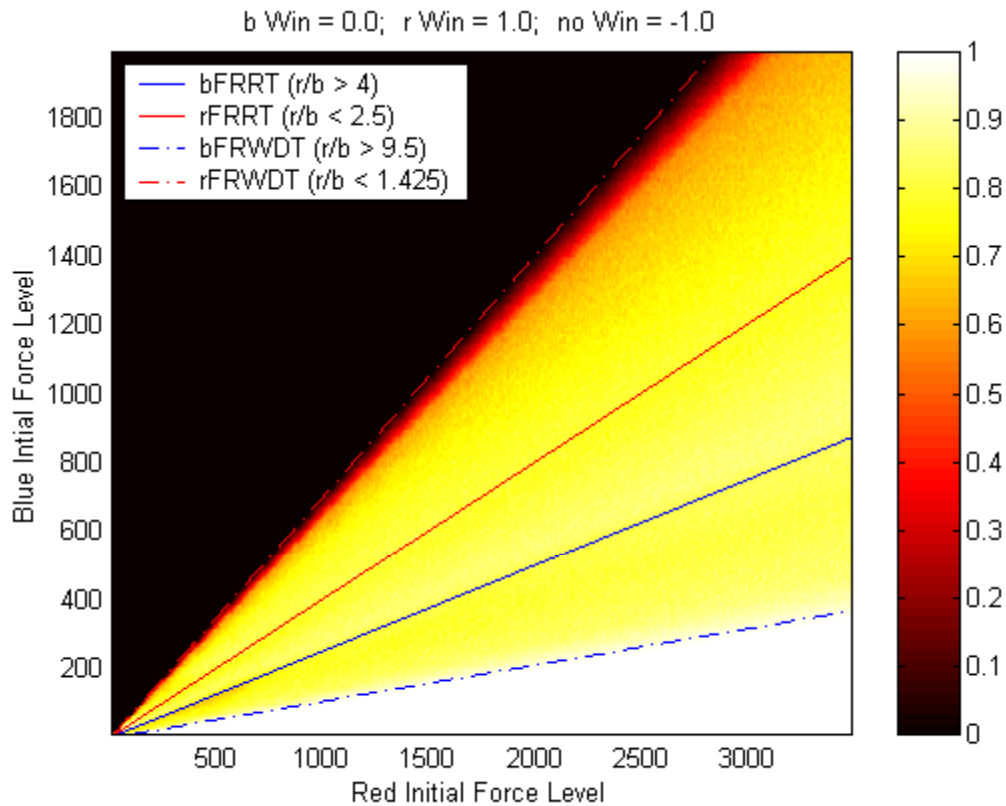


Figure 2.2. An Example of a Stochastic Response Surface.

Just as in Figure 2.1, the reinforcement and withdrawal thresholds are superimposed on the response surface. Note that the region between the withdrawal thresholds has greatly expanded, when compared to Figure 2.2. The clear division between the black region, where Blue always wins, and the white region, where Red always wins, is bounded by the withdrawal thresholds. The red/yellow region, between these thresholds, represents an increasing probability of a Red win, according to the color bar displayed to the right of the graph. This appeals to our intuition: the outcome remains uncertain until one side or the other withdraws.

## **D. CHECKING THE NUMERICS OF THE MODEL**

Julian Palmore (1996) listed three potential problem areas that should be checked in every model to ensure that they are eliminated as a source of instability in the output of a model.

- a. Structurally unstable implementations of branchings on thresholds set by decision tables.
- b. Structural defects in decision logic.
- c. Computer arithmetic.

### **1. Decision Tables and Structural Defects in Decision Logic**

The Dewar model's structure, outlined in Table 2.1, does not contain any branchings on thresholds set by decision tables. Hence, item "a" on Palmore's list of things to check is not a potential source of instability in this model.

The re-coded Java model has the same logical structure as the original Dewar model—i.e., the logical flow through the decision thresholds is identical. First, reinforcement decisions are made, followed by withdrawal decisions. Finally, attrition calculations are made, and then the loop repeats. Cooper (1994) and others have criticized the Dewar model for the lack of realism implied by the structure of the decision thresholds—i.e., the commander makes withdrawal decisions without taking into consideration the number of reserves. This criticism may have merit, but is not an issue in determining whether or not there are defects in decision logic. Based on the fact that the structural decision logic of the re-coded model is identical to the structural decision logic in the original Dewar model and Dewar's analysis of that model, we conclude that the re-coded model is not defective with regard to structural decision logic.

### **2. Computer Arithmetic**

The decisions to reinforce and withdraw, in the Dewar model, are based upon the calculated force ratio (Red/Blue) at each time step. According to Palmore (1996), "Computer arithmetic affects iterative computational processes. Time is measured by the number of iterations performed. Rounding and precision affect outcomes as iteration

proceeds.” The original Dewar model iterated 200,000 times. The re-coded model in Java uses either 10,000 or 20,000 iterations<sup>5</sup>. Therefore, the source with the greatest potential for causing error in the model’s response is the cumulative effect of rounding error due to the “iterative computational processes.”

The single, most active iterative process in the model is the Lanchester Square Law coupled difference equation attrition process. In order to allow this process to continue for an arbitrary number of iterations, a modification of the model is needed. In the Dewar et al. study, a ‘force-ratio only’ model is described and used. This model has no withdrawal threshold and so continues to run until a predetermined number of iterations is reached. The best way to visualize this iterative process is to plot the battle trace—i.e., the force level of each side at each iteration. The effects of rounding error and precision are tested against the attrition process in this ‘force-ratio-only’ model, coded in Java.

### **3. Testing the Effect of Rounding Error and Finite Precision on the Output of the Model**

The Java programming language includes a class called `BigDecimal` that allows variable precision arithmetic. Each of the Blue and Red force levels is represented with both a `BigDecimal` and a ‘double precision’ variable. Java’s ‘double precision’ variables maintain 17 significant digits (precision) in all calculations. Using the variable precision of `BigDecimal`, the number of decimal digits used in a calculation can be specified. The force ratio (Red/Blue) is calculated with both a `BigDecimal` and a ‘double precision’ variable during each iteration. Then, the `BigDecimal` value is converted to a ‘double precision’ value and the two ratios are compared. If they differ by more than one, the model exits. Next, one decimal digit of precision is added to the `BigDecimal` variables, and the model is run again. The return values for each model run are:

- a. Precision, the number of decimal digits of accuracy used by `BigDecimal`.
- b. Number of iterations until the difference between ratios is greater than one.

---

<sup>5</sup> We started this study using only 10,000 iterations. Response surfaces that were non-monotonic with respect to multiple MOEs were subsequently discovered. To maintain focus on the MOE (Who wins), the number of iterations was increased to 20,000.

c. Force Ratios—i.e., BigDecimal (Red/Blue) and double (Red/Blue).

Table 2.2 lists the results of the experiment and the model's parameter settings for each run. As expected, the number of iterations executed, until the force ratios differ by at least one, increases as the precision of the BigDecimal variables increases. Once BigDecimal precision is greater than 17, the number of iterations executed before the force ratios diverge levels off at 3278 iterations. Figure 2.3 is a scatter plot of the results. The graph indicates a strong linear relation between precision and iterations.

Initial Force Levels: 20 (Blue) - 40 (Red)		
Attrition Coefficients: 0.005 (Blue) - 0.02 (Red)		
Force Ratio Reinforcement Thresholds: 4.0 (Blue) - 2.5 (Red)		
Reinforcement Block Size: 10.0 (Blue) - 10.0 (Red)		
P Iters	BigDecimal Ratio	Double Precision Ratio
1 56	3.8	1.8513995374206604
2 56	4.06	1.8513995374206604
3 465	4.045	2.2227715067848046
4 465	4.0605	2.2227715067848046
5 936	4.02194	2.1224803956691356
6 892	2.210905	4.058957745991585
7 1170	1.8846299	4.052567009807578
8 1325	4.05934872	2.192053368452815
9 1776	4.055303309	2.1363901453745644
10 1877	1.8830982468	4.05249263887456
11 2150	2.8410151988	3.8463064329708287
12 2496	2.102862579982	4.051924137329424
13 2569	3.8347182117007	2.8344406281484735
14 3079	2.06417480125275	4.058972776610194
15 3160	4.05678560873031	2.19215081243929
16 3279	3.886759214094909	2.8587189106141215
<b>17 3278</b>	<b>3.842331956483406</b>	<b>2.838151523457638</b>
18 3278	3.8423781034319027	2.838151523457638
19 3278	3.842338402457037	2.838151523457638

Table 2.2. Using Double Precision, Rounding Error Affects the Outcome After Only 3278 Iterations.

BigDecimal is a Java class that allows computation with variable precision real numbers. Double precision variables, in Java, maintain 17 digits of accuracy in all calculations. This table compares the results of simple computations, using these two variables, in order to find the point at which rounding error—an artifact of finite precision—begins to affect the computed results. The row, highlighted in Red indicates the maximum number of iterations possible, when using double precision, before cumulative rounding error affects the first digit to the left of the decimal point.



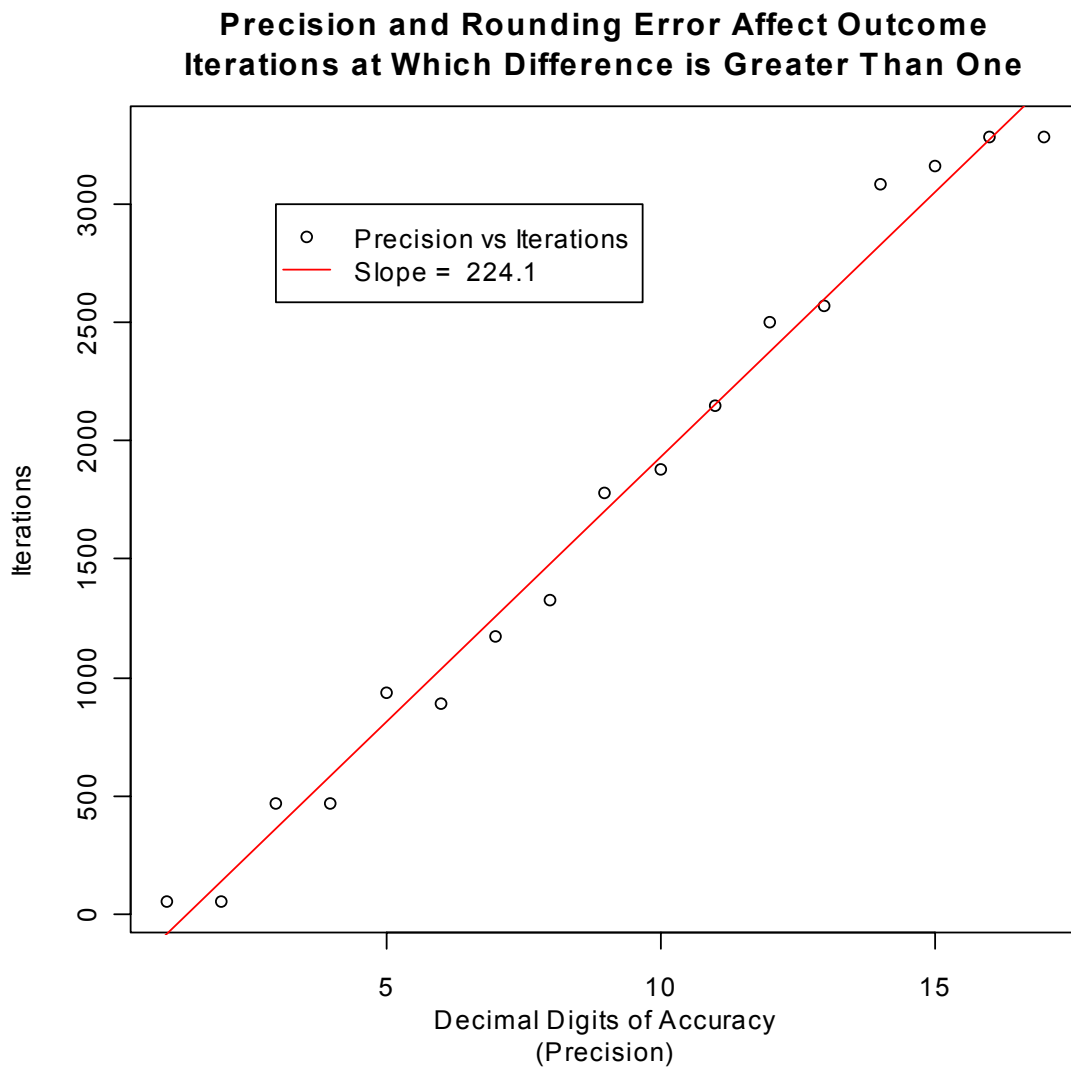


Figure 2.3. How Precision and Rounding Error Affect the Outcome.

This graphs shows that, for each added digit of precision, it takes, on average, another 224 iterations before an iterated computation begins to diverge from the true answer.

#### 4. Results of the Simple Experiment

The results of the experiment imply that, when using ‘double precision’ variables, the calculated orbit—the battle trace calculated using ‘double precision’ variables—differs from the exact orbit by at least one, after only 3278 iterations. A difference of one is enough to change the course of the battle. Let BD be the force ratio

calculated with BigDecimal variables (with precision greater than 17), and DP be the force ratio calculated with ‘double precision’ variables. Then, at the 3278<sup>th</sup> iteration  $|BD - DP| > 1$ . Suppose, at the 3278<sup>th</sup> iterations,  $BD = 3.0$ . Then, either  $DP \leq 2$  or  $DP \geq 4$ . In the first case, ( $DP \leq 2$ ), Red would have reinforced in the ‘double precision’ variable model, while no reinforcement would have occurred in the BigDecimal variable model. In case two, ( $DP \geq 4$ ), Blue would have reinforced in the ‘double precision’ variable model, while no reinforcement would have occurred in the BigDecimal variable model. Therefore, when using Java’s ‘double precision’ variable, the battle trace diverges from the true battle trace after only 3278 iterations. The red line in Figure 2.3—the slope coefficient of ‘iterations’ regressed on ‘precision’—indicates that for each additional digit of precision, it takes approximately another 224 iterations before the calculated orbit begins to diverge from the exact orbit.

The original Dewar model executes 200,000 iterations for each model run. According to the results of the simple experiment, tracking the true battle trace through all these iterations requires approximately  $200,000/224 \cong 893$  digits of precision. The re-coded Java model executes, normally, 20,000 iterations. Therefore, the Java model requires approximately 90 digits of precision to accurately track the true battle trace. Variable precision arithmetic is cumbersome and significantly increases the amount of time needed to run a model. This seems to be a problem. However, a lemma from chaos theory may provide some help.

## 5. The Shadowing Lemma

As mentioned previously, the battle trace of the Dewar model is chaotic. Peitgen et al. (1992, pp 577 – 580) derive the Shadowing Lemma, which, briefly summarized, shows that

[w]hen [comparing] computed orbits with exact orbits then the deviation due to accumulated error propagation will soon amplify so rapidly in the course of the computation that typically any correlation between exact orbits and computed orbits will vaporize. Nevertheless . . . within the shadow of the computed orbit there will be some exact orbit traveling along.

“Within the shadow” is precisely defined in the derivation of the Shadowing Lemma. Given a computed orbit  $\{x_0, x_1, x_2, \dots\}$ , there exist an exact orbit  $\{z_0, z_1, z_2, \dots\}$ , started at some initial point near  $x_0$ , and an  $\varepsilon > 0$  such that  $\varepsilon > |\varepsilon_i| = |z_i - x_i|$  for  $i = 0, 1, 2, \dots$ . The implication here is that, given a chaotic attractor, such as the Dewar battle trace, the effect of accumulated error propagation on the numerical stability of the attractor is bounded by a finite value. Figure 2.4 shows a visual demonstration of the Shadowing Lemma. The Shadowing Lemma implies that, even though the calculated orbit does not follow the exact orbit, point for point, each point in the calculated orbit corresponds to some point or is within  $|\varepsilon|$  of a point in the exact orbit.

The four graphs in Figure 2.4 show the plots of two superimposed battle traces from the ‘force-ratio only’ model. Each battle trace contains 20,000 points, representing the force levels at each of 20,000 iterations. The first battle trace, plotted in each graph of Figure 2.4, is generated using BigDecimal variables, with precision set to 90. According to the results of the simple experiment, using 90 decimal digits of precision for the BigDecimal variables should result in an exact orbit for 20,000 iterations. The second battle trace is generated using ‘double precision’ variables. From the discussion of the simple experiment’s results, ‘double precision’ rounding errors begin to affect the orbit of the battle trace when approximately 3278 iterations have been executed.

The graph in the upper left corner displays the first 3277 points of both battle traces. The BigDecimal battle trace is plotted in black, and then the ‘double precision’ battle trace is plotted in red. As expected, there is no difference in the two orbits. The upper right graph displays the 3278<sup>th</sup> through the 6278<sup>th</sup> points of both battle traces. Again, as expected, each of the points in the two battle traces no longer corresponds to the same orbit. Rounding error has caused the points in the ‘double precision’ battle trace to skip onto, or close to, other points in the orbit. The lower left graph shows the last three thousand points generated by the 20,000 iterations of the model. Again, the points in the ‘double precision’ battle trace and the BigDecimal battle trace are following different orbits. The lower right graph is a plot of the entire battle trace for both the

BigDecimal and the ‘double precision’ variables. At first, it may seem as if the preponderance of the color red is due only to ‘pixel bleed over.’ That is, the resolution of a computer screen or a printer cannot cleanly distinguish between two points that are close together; hence, a point plotted from the first battle trace is overwritten by a nearby—but not equal—point from the second battle trace. Surely this is true for many of the points in Figure 2.4. However, if none of the points were equal, there would be much more black showing than red. To confirm this assertion, Figure 2.5 is an enlarged view of the graph in the lower right corner of Figure 2.4. Note the preponderance of the color red, an effect implied by the Shadowing Lemma.

Peitgen et al. emphasize two implications of the Shadowing Lemma:

[First,] . . . the shadowing lemma should not mislead us to think that it provides us with a way to escape the consequences of sensitivity to initial condition[s] . . . [Second,] . . . the shadowing lemma does ensure us that statistical properties measured by computer experiments are in fact significant.

We conclude from the foregoing that the chaotic battle trace remains chaotic even in the presence of cumulative rounding error and the finite precision of Java’s ‘double precision’ variables. Now that we have looked at the stochastic version of the Dewar model, re-coded in Java, we turn to the problem of searching for additional non-monotonic subspaces.

The Shadow Lemma and Numerical Stability  
Battle Trace of Dewar Model

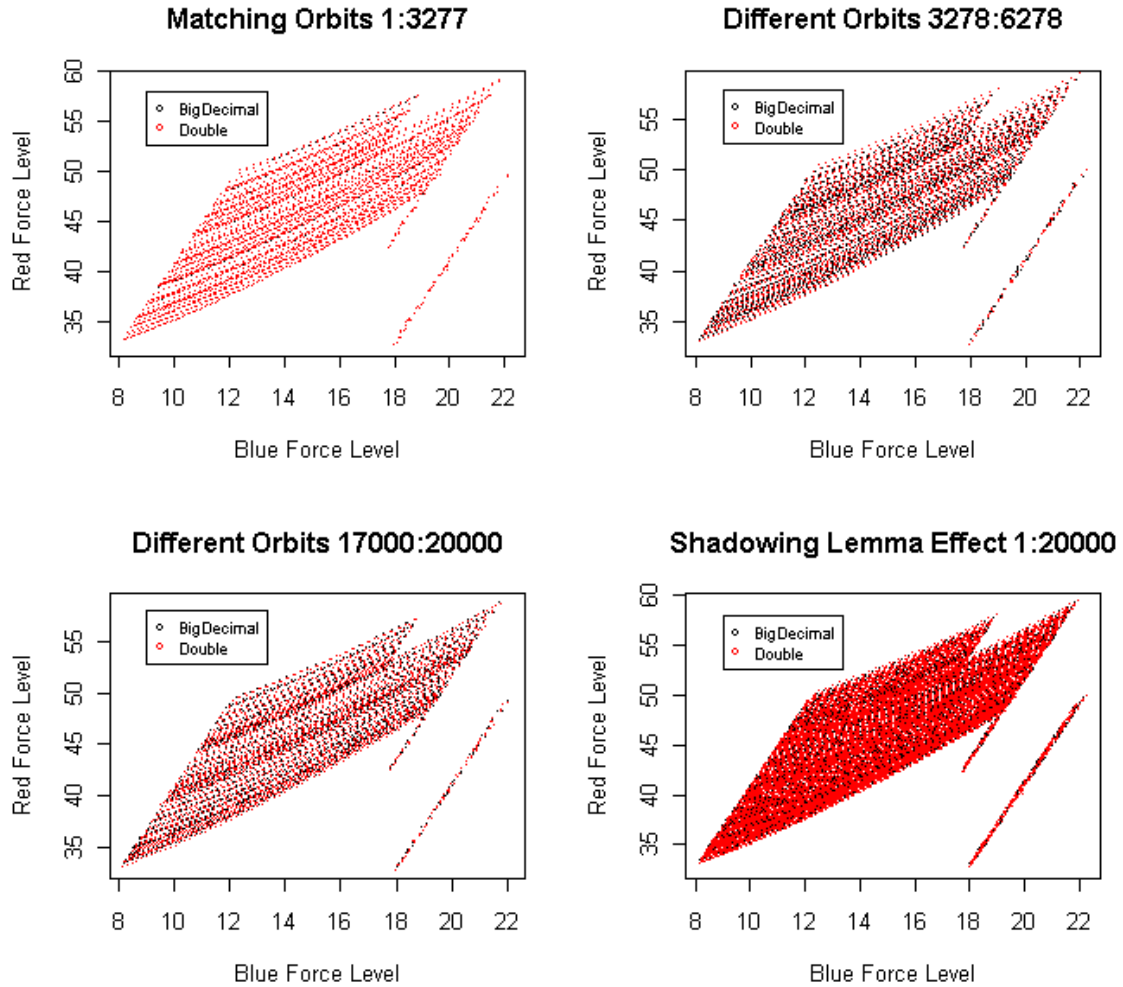


Figure 2.4. Shadowing Lemma Effect and Numerical Stability.

These graphs illustrate the Shadowing Lemma, a result from chaos theory that implies that chaotic attractors are stable in the presence of cumulative rounding error. Note that, even though the calculated battle trace begins to diverge from the exact battle trace after only 3278 iterated computations, the calculated battle is always within the *shadow* of the exact battle trace.

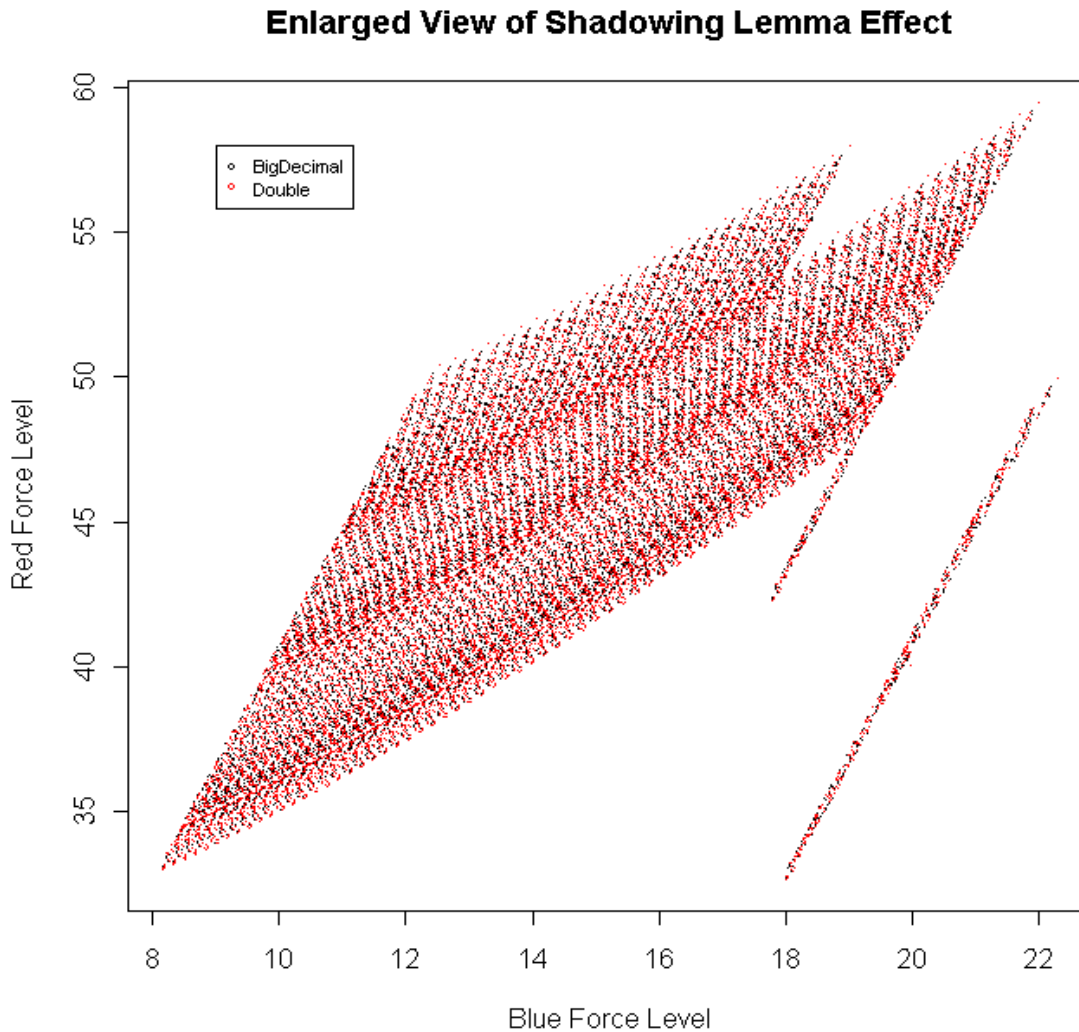


Figure 2.5. Enlarged View of the Shadowing Lemma Effect.

This graph is an enlarged view of the graph in the lower right corner of Figure 2.4. Two battle traces have been superimposed on the same graph. The first was calculated using arbitrary precision (BigDecimal). Hence, it represents the exact battle trace. The second was calculated using double precision (17 digits of precision). Each of the battle traces contains 20,000 points. This enlarged view shows, more clearly, that the second battle trace remains in the *shadow* of the first—i.e., cumulative rounding error, an artifact of finite precision, does not cause numerical instability; the double precision battle trace is, in a sense, just as accurate as the exact battle trace. Therefore, the double precision battle trace remains sensitive to initial conditions.

THIS PAGE INTENTIONALLY LEFT BLANK

### **III. LATIN HYPERCUBE SAMPLING: SEARCHING FOR NON-MONOTONICITY**

This chapter briefly describes the Latin Hypercube Sampling method. The design for the sample is developed, and the samples are taken. The results of the sampling suggest that non-monotonicity is more pervasive in the Dewar model than previously thought. The conclusions of this chapter lead to the need for a fractional factorial experiment, defined and designed in the next chapter.

#### **A. LATIN HYPERCUBE SAMPLING DESIGN**

As outlined in Chapter II, the Dewar model has an 18 dimensional phase space. Dewar et al. found significant non-monotonicity in the Initial Force Level subspace. The question remains: are there other subspaces that exhibit significant non-monotonic behavior when additional capability is added to only one side? In order to answer this question, a systematic look across the entire phase space is necessary. Because the combinatorics prohibit a full factorial search, a more efficient way to conduct this search must be found.

In 1979, McKay et al. published a method called Latin Hypercube Sampling (LHS). This method provides a more thorough “space-filling” search and “can be viewed as a K-dimensional extension of Latin square sampling.” In 1998, Ye developed a variant of the LHS method, Orthogonal LHS (OLHS). However, the designs allowed by the OLHS method are fairly restrictive; you lose orthogonality unless you restrict yourself to a fairly small subset of the possible designs. We use the LHS method vice the OLHS method for two reasons. First, we do not want to restrict the search to the narrow confines of an OLHS design. Second, the primary goal is to determine if additional non-monotonic behavior is present in the Dewar Model, rather than to calculate the effects of changes in the variables. Therefore, we give up orthogonality in favor of an increased capability to search a broader region of the 18 dimensional space of the Dewar model.



## 1. Implementing the Latin Hypercube Design

The LHS method generates experimental design points in the following manner. A single variable is chosen as the ‘free’ variable. This free variable is assigned a sequence of values over which it will vary for each experimental run. Let  $n$  be the number of free variable values in the assigned sequence. Each remaining variable is assigned an interval that is divided into equal subintervals. In a ‘main effects’ Latin Hypercube, as we run here, the number of subintervals is equal to the number of remaining variables. In the case of the Dewar model, the assigned interval of each remaining variable is divided into 17 subintervals. The midpoints of each of these subintervals, together with the  $n$  values of the ‘free variable,’ become the elements of one design point,  $X$ , i.e.,

$$X = \{fv\} \cup [x_1, x_2, \dots, x_{17}],$$

where  $fv$  is the sequence of free variable values and each  $x_i$  is a midpoint from one of the subintervals of the 17 remaining variables. For each experimental run, one midpoint is randomly chosen from a subinterval of each of the remaining variables; any given midpoint is chosen only once. Just as in a Latin Square design, each 18-tuple combination of midpoints occurs only once. This ensures that the broadest possible range of the entire phase space is sampled. Using this method, the number of experimental design points is equal to the number of free variables times the number of remaining variables—i.e.,  $n(n-1)$ . In the case of the Dewar model, an LHS experiment designed in this manner results in  $18 \times 17 = 306$  experimental design points. This equates to  $306 \times 69451$ , or 21,252,006 model runs. However, due to the symmetry of the Dewar model, the number of runs can be reduced by more than half.

## 2. Modifying the LHS Design

Notice that the Dewar model variables come in symmetric pairs—e.g., Blue and Red Initial Force Levels; Blue and Red Attrition Coefficients; etc. Table 3.1 shows this pairing. It is natural to pair these parameters as the ‘free variables’ and allow them to vary together. Thus, the output of each design point will be a response surface rather than a curve. Since we use two free variables at a time, there are nine pairs of free

variables and 16 remaining variables after each pair of free variables is chosen. Thus, Therefore, the number of experimental design points is reduced from 306 to  $9 \cdot 16 = 144$ . Thus, only 10,000,944 model runs are necessary. However, no information is lost. In fact, we end up exploring a greater range of the model's 18 dimensions this way. If only one free variable is used each time, each of the remaining variables is allowed only 17 distinct values. By allowing each of the two variables to vary across a much broader region of its domain, the breadth of the search is dramatically increased. Of course, there are many other possibilities.

Instead of using only symmetric pairs, we could use each possible combination of pairs of variables. This would result in the need for  $\binom{18}{2} \cdot 16 = 2448$  experimental runs.

The advantage, of course, would be the ability to visualize two-way interactions between each of the variables. This idea could be expanded to include three-way interactions by allowing three free variables at a time. However, the number of required runs quickly becomes unrealistic, even if 2448 runs don't already seem excessive. Since the primary goal of the LHS experiment is to search for non-monotonicity rather than to evaluate interactions, using symmetric pairs of the variables seems to be a reasonable choice.

### **3. How Thick Should the Hyperplanes Be?**

Another point requires careful consideration. When designing an LHS experiment, how large should the domains of the variables be? Should the hyperplanes be thick or thin? The answer depends on the model and what makes sense with regard to the values of the variables. The original Dewar model was structured to represent a smaller, more efficient force opposing a larger, less effective force or, if you prefer, a smaller Blue defensive force opposing a larger Red attacking force. If the domains of free variables and the remaining variables are allowed to range over too wide an interval, some of the design points will be meaningless, just an agglomeration of random numbers. On the one hand, since we want to preserve the original model's basic structure and tension between opposing forces, we restrict the domain of the remaining variables to fairly thin hyperplanes, centered around the nominal values of the original model. On the

other hand, we allow the free variables to vary over a much wider interval centered on the nominal values in the original Dewar model<sup>6</sup>. Using this reasoning, the resulting response surfaces must be carefully inspected so that inappropriate regions of the response surface are not used to draw conclusions about the model's behavior.

Once the basic structure is determined, the experiment is designed and implemented by stepping through each of the free variable pairs, calculating the input values for each design point and then running the model for each design point. Table 3.1 shows the design used for this paper.

Variable Names	Original Dewar Model Values	Free Variable Values			Remaining Variable Values		Non-monotonic Response Surfaces
		Lower Bound	Upper Bound	Step Size	Lower Bound	Upper Bound	
<b>bIF</b>	40.00	10.00	2000.00	10.00	300.00	600.00	<b>0-15</b>
<b>rIF</b>	80.00	10.00	3500.00	10.0	800.00	1700.00	
<b>bFRRT</b>	4.00	3.00	5.00	0.0100	3.950	4.050	<b>0, 3-10, 12-15</b>
<b>rFRRT</b>	2.50	1.50	3.50	0.0100	2.450	2.550	
<b>bPIFRT</b>	0.80	0.70	0.90	0.00	0.7950	0.805	<b>5, 6, 9</b>
<b>rPIFRT</b>	0.80	0.70	0.90	0.00	0.7950	0.805	
<b>bFRWDT</b>	10.00	9.00	11.00	0.0100	9.950	10.050	
<b>rFRWDT</b>	1.50	0.50	2.50	0.0100	1.450	1.550	
<b>bPIFWDT</b>	0.70	0.60	0.80	0.00	0.6950	0.7050	
<b>rPIFWDT</b>	0.70	0.60	0.80	0.00	0.6950	0.7050	
<b>bRBA</b>	6.00	1.00	200.00	1.00	5.00	6.00	<b>1</b>
<b>rRBA</b>	6.00	1.00	200.00	1.00	5.00	6.00	
<b>bRBD</b>	70.00	0.00	100.00	0.50	69.00	71.00	<b>0-15</b>
<b>rRBD</b>	70.00	0.00	100.00	0.50	69.00	71.00	
<b>bRBS</b>	300.00	10.00	500.00	2.4500	290.00	300.00	<b>0-6, 9-15</b>
<b>rRBS</b>	300.00	10.00	500.00	2.4500	290.00	300.00	
<b>bAC</b>	1/2048	0.000010	0.0010	0.000004950	0.000478281250	0.000498281250	<b>0-13, 15</b>
<b>rAC</b>	1/512	0.000020	0.0020	0.000009900	0.001852125000	0.002053125000	

Table 3.1. Latin Hypercube Sampling Experiment Design and Results.

78 of the 144 response surfaces sampled, exhibited non-monotonic behavior. Additionally, many of the surfaces were non-monotonic with respect to multiple MOEs.

<sup>6</sup> Note that two of the parameters are integer valued (RBA, RBD). When their values are perturbed by 5 percent, 16 distinct values are not created.

#### **4. Results of the Latin Hypercube Design Experiment**

The first cell in the last column corresponds to the initial forces subspace, where the response surfaces, generated by the sixteen design points, all exhibited non-monotonic trends. In some cases, there was more extreme non-monotonicity than in the original Dewar model. The red numbers in each cell indicate the design points that resulted in non-monotonic response surfaces. This table indicates that non-monotonicity is pervasive in at least six of the two-dimensional subspaces of the deterministic Dewar model. There are 152 other two-dimensional subspaces of the Dewar model that were not searched. It is possible that additional non-monotonic behavior could be found in these subspaces as well.

Not only was non-monotonicity lurking in the suburbs and shadows, but it was also parading down prominent avenues of the Dewar model's hyperspace—see the graphs in Figure 3.1 and the last column of Table 3.1. The graph in the lower right corner of Figure 3.1 shows the 14<sup>th</sup> experimental run of the Blue and Red Reinforcement-Block-Size (RBS) pair of free variables. This graph is a projection of the other sixteen dimensions onto the RBS plane. Dewar et al. used several different RBS values as they looked at the Initial Force Level (IF) subspace. However, the non-monotonicity displayed here was not previously discovered. Similarly, the other graphs in Figure 3.1 are representative of the scores of non-monotonic surfaces newly discovered in distinct two-dimensional subspaces of the Dewar model's phase space.

#### **5. Non-monotonicity in a Different Measure of Performance**

The measure of effectiveness (MOE) we focus on—the same used by Dewar, et al.—is: “who wins?” Two of the many striking examples we discovered, using Latin Hypercube Sampling, are displayed in the top row of Figure 3.1. These two graphs exhibit non-monotonicity with respect to a second MOE “length of battle.” The two bottom graphs in Figure 3.1 exhibit non-monotonicity with respect to the MOE “who wins?” All of these graphs represent projections of the 18 dimensions of the Dewar model onto the two-dimensional subspaces indicated by the graph titles. Each of these response surfaces is generated by the deterministic Dewar model.

## Two-Dimensional, Non-monotonic Subspaces

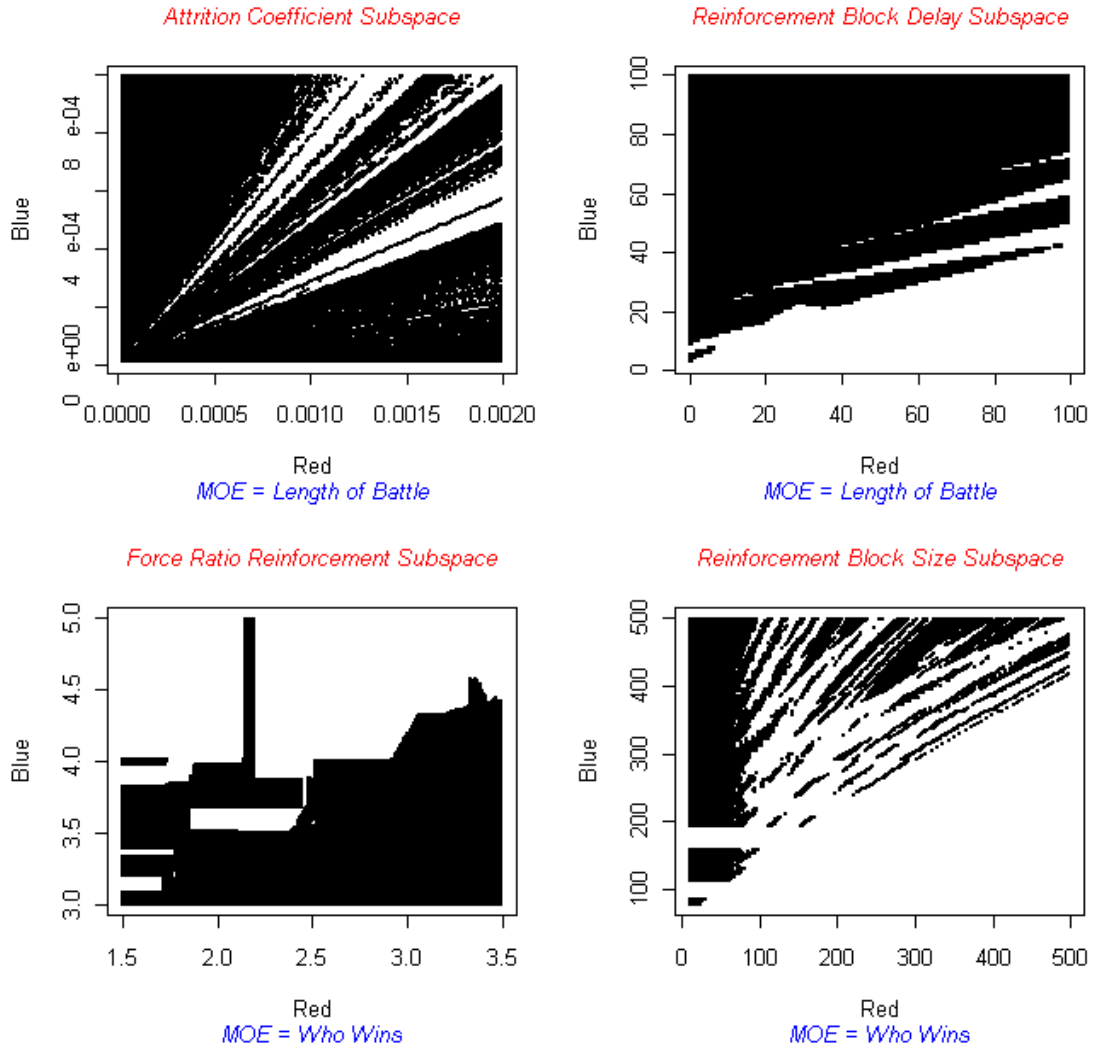


Figure 3.1. Non-monotonicity Appears Throughout the Original Dewar Model. The top two graphs show non-monotonic behavior with respect to the MOE “Length of Battle,” i.e., the battle was still going after 10,000 iterations. The two graphs in the bottom row exhibit non-monotonic behavior with respect to the MOE “who wins?”

In our implementation of the Dewar Model for the LHS experiment, the battles run for only 10,000 time steps. If neither side wins after 10,000 time steps, the model returns “-1.0.” The black regions in the graphs in the upper right and left corners of Figure 3.1 exhibit non-monotonic behavior with respect to the MOE “length of battle,” in the Attrition Coefficient and Reinforcement Block Delay subspaces. The black regions

in the graphs in the lower left and right corners of Figure 3.1 exhibit non-monotonic behavior with respect to the MOE “who wins?” in the Force Ratio Reinforcement and Reinforcement Block Size subspaces. The pervasiveness of significant non-monotonicity in this simple model would seem to make its results highly suspect. Is there any way to control, if not eliminate, this non-monotonicity?

## **6. Can Non-monotonicity Be Reduced or Eliminated?**

Dewar et al. discovered that the non-monotonic behavior in their simple model was due to the chaotic behavior induced by the forcing and damping behavior of the reinforcement thresholds and the attrition equations, respectively. Scientists in many diverse fields have discovered that it is possible to control chaotic behavior in some systems, to varying degrees, by randomly perturbing their systems, (Gleick, 1987). Additionally, during the past nine years, advances in computer technology have given analysts a much-improved capability to explore solutions to the problems highlighted by the Dewar study. One such method, proposed in various forms by Allen (1993), Louer (1994), Lucas (1997) and Seager (2000), is stochastic perturbation of the input variables, specifically those variables used in the decision thresholds and the attrition coefficients.

This idea is implemented with the Dewar model. We stochastically perturb various input parameters to see what effect they have on the response surfaces, both those discovered by Dewar et al. and those we found using the LHS experiment. The intent is to take a systematic, in-depth look at the effect of stochastic perturbations on the response surface. The results may be useful in helping determine which of the 18 parameters should be made stochastic, if any. A fractional factorial experimental design can accomplish this goal in an economical way. The next section addresses this experimental design.

THIS PAGE INTENTIONALLY LEFT BLANK

## IV. THE FRACTIONAL FACTORIAL EXPERIMENT

### A. DETERMINING THE EFFECTS OF STOCHASTIC MODELING ON THE MODEL'S RESPONSE SURFACES

The results of the Latin Hypercube Sampling reveal that six of the nine paired parameters produce non-monotonic response surfaces (see Table 3.1, “Non-monotonic Runs” column). Rather than try to analyze these six subspaces simultaneously, we run a fractional factorial design experiment on the initial force level subspace, the same subspace examined by Dewar et al. There are two constraints for each of the reinforcement and withdrawal thresholds; these eight constraints plus the other ten parameters give us the 18 dimensions of the Dewar model (The original Dewar model parameters are shown in Table 2.1).

Based on the results of the Latin Hypercube Sampling, we choose a  $2^{9-3}$  resolution V design with the following generators:  $I = 123457, 123468, 123569$ . This design ensures that no main effect is confounded with any 2<sup>nd</sup>, 3<sup>rd</sup> or 4<sup>th</sup> order interaction. It also ensures that 2<sup>nd</sup> order interactions are not confounded with 3<sup>rd</sup> order interactions or other 2<sup>nd</sup> order interactions (Box et al., 1978). Since the primary focus is on the main effects, the first six columns are written out as a complete  $2^6$  factorial design. The subsequent columns are constructed using the generators listed above. The order of the runs is not randomized since their order will not have any effect on the responses.

The design in Table 4.1 is the result of the considerations listed above. The “1” and “-1” correspond to “stochastic” and “not stochastic,” respectively. The attrition coefficients, when stochastic, are modeled as normal random variates with means equal to the nominal Dewar model values and standard deviations equal to ten percent of the square root of their means. All other parameters, when stochastic, are modeled as uniform random variates with intervals centered on the nominal Dewar model values and ranging plus and minus five percent of the nominal value. These distributions and values are based on the work done by Lucas (1997) and Seager (2000). In order to evaluate the



experimental results, we need parameters that measure the non-monotonicity of the response surface. These parameters are the subject of the next chapter.

Generators: I = 123457, 123468, 123569																			
Thesis Fractional Factorial Design (64)																			
Run	1	2	3	4	5	6	7	8	9	Run	1	2	3	4	5	6	7	8	9
	IF	FRRT	PIFRT	RBD	RBS	AC	FRWDT	PIFWDT	RBA		IF	FRRT	PIFRT	RBD	RBS	AC	FRWDT	PIFWDT	RBA
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	33	-1	-1	-1	-1	-1	1	-1	1	1
2	1	-1	-1	-1	-1	-1	1	1	1	34	1	-1	-1	-1	-1	1	1	-1	-1
3	-1	1	-1	-1	-1	-1	1	1	1	35	-1	1	-1	-1	-1	1	1	-1	-1
4	1	1	-1	-1	-1	-1	-1	-1	-1	36	1	1	-1	-1	-1	1	-1	1	1
5	-1	-1	1	-1	-1	-1	1	1	1	37	-1	-1	1	-1	-1	1	1	-1	-1
6	1	-1	1	-1	-1	-1	-1	-1	-1	38	1	-1	1	-1	-1	1	-1	1	1
7	-1	1	1	-1	-1	-1	-1	-1	-1	39	-1	1	1	-1	-1	1	-1	1	1
8	1	1	1	-1	-1	-1	1	1	1	40	1	1	1	-1	-1	1	1	-1	-1
9	-1	-1	-1	1	-1	-1	1	1	-1	41	-1	-1	-1	1	-1	1	1	-1	1
10	1	-1	-1	1	-1	-1	-1	-1	1	42	1	-1	-1	1	-1	1	-1	1	-1
11	-1	1	-1	1	-1	-1	-1	-1	1	43	-1	1	-1	1	-1	1	-1	1	-1
12	1	1	-1	1	-1	-1	1	1	-1	44	1	1	-1	1	-1	1	1	-1	1
13	-1	-1	1	1	-1	-1	-1	-1	1	45	-1	-1	1	1	-1	1	-1	1	-1
14	1	-1	1	1	-1	-1	1	1	-1	46	1	-1	1	1	-1	1	1	-1	1
15	-1	1	1	1	-1	-1	1	1	-1	47	-1	1	1	1	-1	1	1	-1	1
16	1	1	1	1	-1	-1	-1	-1	1	48	1	1	1	1	-1	1	-1	1	-1
17	-1	-1	-1	-1	1	-1	1	-1	1	49	-1	-1	-1	-1	1	1	1	1	-1
18	1	-1	-1	-1	1	-1	-1	1	-1	50	1	-1	-1	-1	1	1	-1	-1	1
19	-1	1	-1	-1	1	-1	-1	1	-1	51	-1	1	-1	-1	1	1	-1	-1	1
20	1	1	-1	-1	1	-1	1	-1	1	52	1	1	-1	-1	1	1	1	1	-1
21	-1	-1	1	-1	1	-1	-1	1	-1	53	-1	-1	1	-1	1	1	-1	-1	1
22	1	-1	1	-1	1	-1	1	-1	1	54	1	-1	1	-1	1	1	1	1	-1
23	-1	1	1	-1	1	-1	1	-1	1	55	-1	1	1	-1	1	1	1	1	-1
24	1	1	1	-1	1	-1	-1	1	-1	56	1	1	1	-1	1	1	-1	-1	1
25	-1	-1	-1	1	1	-1	-1	1	1	57	-1	-1	-1	1	1	1	-1	-1	-1
26	1	-1	-1	1	1	-1	1	-1	-1	58	1	-1	-1	1	1	1	1	1	1
27	-1	1	-1	1	1	-1	1	-1	-1	59	-1	1	-1	1	1	1	1	1	1
28	1	1	-1	1	1	-1	-1	1	1	60	1	1	-1	1	1	1	-1	-1	-1
29	-1	-1	1	1	1	-1	1	-1	-1	61	-1	-1	1	1	1	1	1	1	1
30	1	-1	1	1	1	-1	-1	1	1	62	1	-1	1	1	1	1	-1	-1	-1
31	-1	1	1	1	1	-1	-1	1	1	63	-1	1	1	1	1	1	-1	-1	-1
32	1	1	1	1	1	-1	1	-1	-1	64	1	1	1	1	1	1	1	1	1

Table 4.1.  $2^{9-3}$  Resolution V Fractional Factorial Experimental Design.

## **V.     DEFINING NON-MONOTONICITY PARAMETERS**

### **A.     THE NEED FOR PARAMETERS TO MEASURE NON-MONOTONICITY**

Except for a single deterministic run in which none of the model parameters are stochastic, all of the fractional factorial experimental design runs have at least one stochastic parameter. Therefore, the response surface will exhibit random fluctuations—i.e., some percentage of the points along the response surface will exhibit non-monotonic behavior due only to the randomness we have added to the model. In order to analyze the response surfaces generated by the stochastic version of the model, we must be able to tell the difference between truly non-monotonic response surfaces and those surfaces whose non-monotonicity is merely the result of the random fluctuations due to the stochastic parameters. The problem is how to separate these random fluctuations from bona fide non-monotonicities—that is, non-monotonicities that will be present as the sample size goes to infinity. To be able to distinguish between random non-monotonicities and real non-monotonicities, we need to find a reasonable bound for the magnitude of the random fluctuations. Any non-monotonicities larger than this bound are probably due to some attribute of the model other than the randomness that we have added to it. In other words, this bound can be used to discriminate between random noise and any truly non-monotonic behavior manifesting itself in the response surface.

In addition to being able to discriminate between random noise and real non-monotonicities, we need to know how many real non-monotonicities there are, and if there are any non-monotonic trends. Non-monotonic trends are two or more successive non-monotonic points. Thus, we need parameters that measure these attributes of the non-monotonicity or roughness of the response surfaces. Ideally, these parameters will be small when the surface is monotonic and will increase in magnitude as the surface becomes increasingly non-monotonic. We turn now to a discussion of the distribution of the points on the response surface generated by the stochastic version of the Dewar model. Understanding how these points are distributed will help us define the non-monotonicity parameters.

The outcome of a single replication of the model is, in essence, the outcome of a Bernoulli trial. A success is denoted by a “1.0,” in which case Red wins. A failure is denoted by a “0.0,” in which case Red loses (i.e., Blue wins). In the fractional factorial experiment, the model is replicated 1000 times for each point. Thus, each point on the response surface is the mean of 1000 replications. Since a Red win is defined as a success, the mean of 1000 replications equates to the proportion of the 1000 replicated outcomes that Red wins. Let  $p$  be the true proportion of Red successes. Let

$\hat{p} = \frac{\# \text{successes}}{n}$ , be the estimated probability that Red wins, given  $p$ . The Central Limit

Theorem implies that the distribution of  $\hat{p}$  is approximately normal, provided the number of replications is large enough to overcome the natural skewness of the Bernoulli distribution. If the number of trials,  $n$ , times the probability of success,  $p$ , and  $1 - p$  is greater than five, i.e.,  $np > 5$  and  $n(1 - p) > 5$ , then the normal approximation is accurate (Devore, 1995). This implies that the normal approximation can be used, provided that the response surface does not fall below 0.005 or go above 0.995. If the response surface falls below or above these respective values and the response surface is fairly smooth, there is no practical ambiguity with regard to the outcome. However, if the response surface falls above or below these respective values and the surface is highly non-monotonic, there will be practical ambiguity with regard to the outcome. If the non-monotonicity parameters are properly defined and precisely calculated, then they should be fairly sensitive to these extreme cases.

The standard deviation of the true population parameter is  $\sigma_{\hat{p}} = \sqrt{\frac{p(1-p)}{n}}$ .

The variance of  $\hat{p}$  is greatest when the probability of success is 0.5. Figure 5.1 shows the standard deviation as a function of the probability of success.

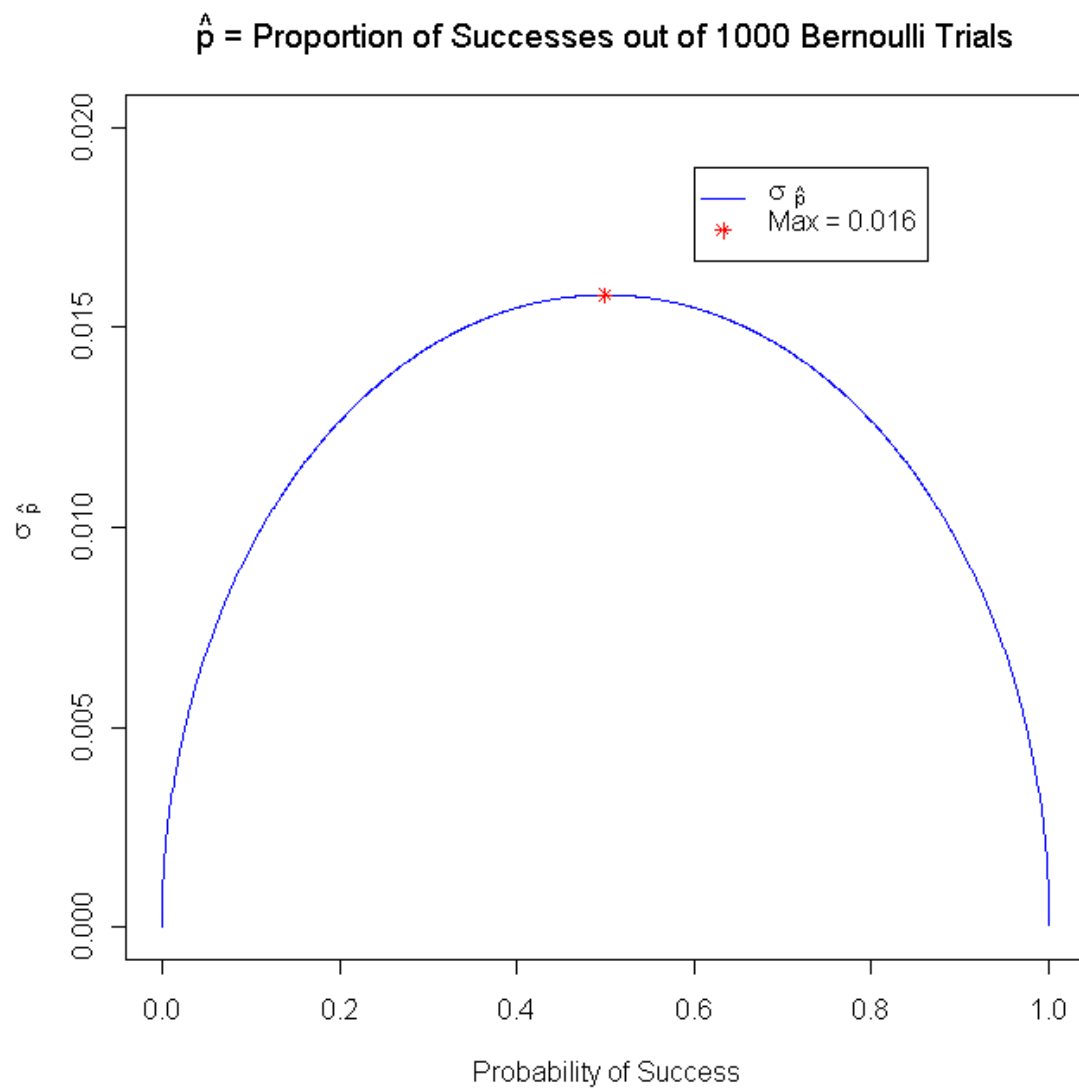


Figure 5.1. Standard Deviation of Response Surface Points When  $n = 1000$ .

Each response surface point is approximately normally distributed. The variance of these points is greatest when the probability of success is 0.5. Compare this Figure with Figure 5.2.

Each point on the response surface is generated independently of the other points. Thus, all of the points on the response surface between 0.005 and 0.995 are independent and, approximately, normally distributed. Since we know the distribution of the points on the response surface, we can use this information to define parameters with which to measure the non-monotonicity of the response surfaces generated by the model.

Given a response surface, we can now determine whether or not it is ‘monotonic’<sup>7</sup> by using the following procedure. Check for non-monotonic behavior by holding one of Blue ( $x$ ) or Red ( $y$ ) constant and numerically "observe" the response surface values that lie along the chosen Blue or Red contour. For example, suppose that the two-dimensional subspace variables are initial force levels. Fix the Red initial force level,  $y$ , at 500, and examine the projection of the response surface onto the  $x$ - $z$  plane. This projection is a curve in the  $x$ - $z$  plane.

The sign of a simple linear regression slope coefficient— $z$  regressed on  $x$  in this case—determines whether the overall trend of the response curve, along the given contour (in this case  $y = 500$ ), in the  $x$ - $z$  plane is positive or negative (increasing or decreasing). Suppose that the regression coefficient is positive—i.e., increasing probability of Red win as initial Red force level increases. If the response surface is truly monotonic, we expect each successive point on the response curve to be at least as big as its immediate predecessor. Of course, due to random variation, there may be some small non-monotonicities when the limiting underlying curve is monotonic. But if the underlying curve is truly monotonic, then the magnitude of these non-monotonicities, due to random variation, will infrequently fall more than three standard deviations above or below the limiting underlying curve.

Given that the overall trend of the response surface is increasing, we construe successive points as monotonic as long as they are at least as big as their immediate predecessors. Those points that fall below their immediate predecessor we call non-monotonic. But this non-monotonicity may be due only to the random fluctuations of the stochastic parameters in the model. If this is not the case—suppose that the next

---

<sup>7</sup> The single quotes are used here to distinguish between a ‘monotonic’ stochastic surface, as described in this chapter, and a response surface that meets the strict mathematical definition of a monotonic non-decreasing (non-increasing) surface.

point falls below the preceding point by three or more standard deviations—then we construe the successive point as a statistically significant non-monotonicity.

Comparing successive points along the response curve is the same as finding the difference between the values of successive points along the response curve. More specifically, let  $p_1$  and  $p_2$  be the true proportions of successes (Red wins) at two points on the response surface. Also, let  $\hat{p}_1$  and  $\hat{p}_2$  be the empirical estimates of  $p_1$  and  $p_2$ . As shown above,  $\hat{p}_1$  and  $\hat{p}_2$  are independent, approximately, normally distributed random variables. Their difference  $Y = \hat{p}_2 - \hat{p}_1$  is also a normal random variable, where

$$E[Y] = E[\hat{p}_2 - \hat{p}_1] = E[\hat{p}_2] - E[\hat{p}_1] = p_2 - p_1, \text{ and}$$

$$Var[Y] = Var[\hat{p}_2 - \hat{p}_1] = Var[\hat{p}_2] + Var[\hat{p}_1] = \frac{p_2(1-p_2) + p_1(1-p_1)}{n}.$$

Therefore, when  $\hat{p}_1$  and  $\hat{p}_2 \in (0.005, 0.995)$ , the distribution of Y is:

$$Y \approx Normal\left(p_2 - p_1, \frac{p_2(1-p_2) + p_1(1-p_1)}{n}\right).$$

Note that the standard deviation of Y,  $\sigma_Y$ , is larger than  $\sigma_{\hat{p}}$ , as can be seen by comparing Figure 5.1 with Figure 5.2. Additionally, note that the greatest variance for Y occurs when  $p_1$  and  $p_2$  are both 0.5. Now that we know the distribution of the difference between points along a contour of the response surface, we can test the hypothesis that the magnitude of some non-monotonicity is significant. For an increasing surface, let the null hypothesis be,  $H_o : p_2 - p_1 \geq 0$ , i.e., the response curve is monotonic non-decreasing. The alternative hypothesis is:  $H_a : p_2 - p_1 < 0$ . Then, calculate the test statistic

$$z = \left| \frac{(\hat{p}_2 - \hat{p}_1)\sqrt{1000}}{\sqrt{\hat{p}_2(1-\hat{p}_2) + \hat{p}_1(1-\hat{p}_1)}} \right|,$$

Since we care only about non-monotonic points, we calculate this statistic only for non-monotonic deviations from the overall trend of the response surface.

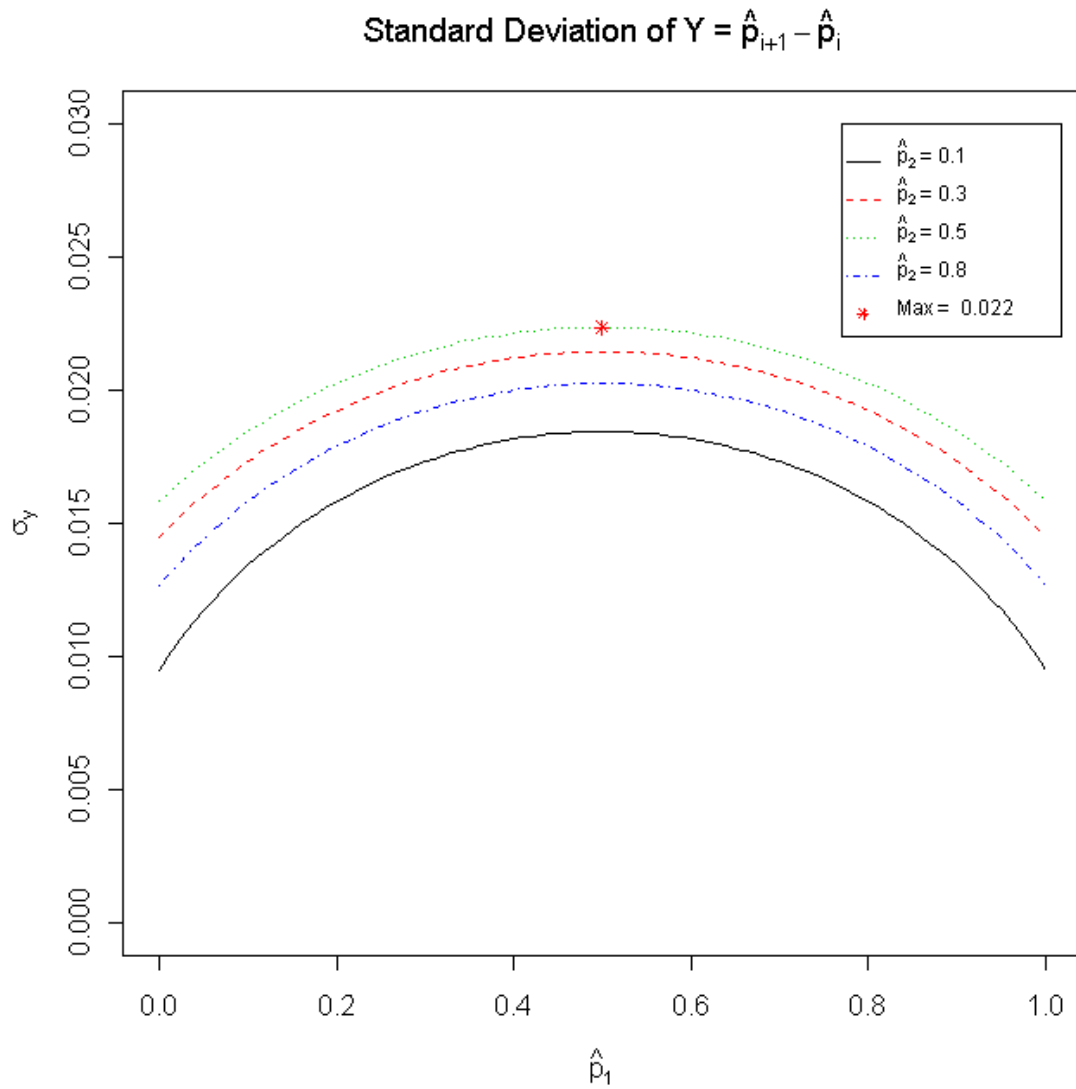


Figure 5.2. Standard Deviation of  $Y = (\hat{p}_{i+1} - \hat{p}_i)$ .

This graph indicates that the maximum variance in the value of  $Y$  occurs when  $\hat{p}_{i+1}$  and  $\hat{p}_i$  are both equal to 0.5. Compare the maximum variance, listed in the legend of this Figure, with the maximum variance of  $\hat{p}_i$  in Figure 5.1.  $\hat{p}_i$  and  $Y$  are both approximately normally distributed, but the variance of  $Y$  is about thirty percent greater than the variance of  $\hat{p}_i$ . We expect this increase in variance, as variance is additive, even when taking the difference between two random variates.

More precisely, let  $p = f(x)$  be the response curve and let  $r = \frac{\sum xp - n\bar{x}\bar{p}}{\sum x^2 - n\bar{x}^2}$  be the slope coefficient of a simple linear regression of  $p$  on  $x$  (Devore 1995). If  $r > 0$  and  $p[i+1] - p[i] < 0$ , or  $r < 0$  and  $p[i+1] - p[i] > 0$ , then  $p[i+1]$  is a non-monotonic deviation from the overall trend of the response surface. If  $z \geq 3$ , then  $H_0$  is rejected and  $p[i+1]$  is a statistically significant non-monotonic deviation from the overall trend of the response surface.

Thus, we have found a potential bound for the magnitude of the random fluctuations in the response surfaces generated by the stochastic version of the Dewar model. Even though non-monotonic deviations from the trend of the response surface greater than two standard deviations are likely to be caused by something other than the random fluctuations of the stochastic parameters, we want to increase the likelihood that non-monotonic deviations in the response surface are truly non-monotonicities and not just random fluctuations due to the stochastic parameters. Therefore, three standard deviations are used as the boundary instead of two. For a standardized normal random variate, we expect to see observations three standard deviations from the mean about 1.3 times in a thousand.

However, the situation is further complicated by the fact that the difference of each successive pair of points along the response surface is not independent—i.e., two successive comparisons share a common point. Thus, there is dependence among the multiple comparisons. This makes finding the exact sampling distribution and error rates, analytically, very difficult. Instead, the sampling distribution and error rates are determined numerically. First, however, we give a brief description of the program that will generate the distribution.

## **B. THE NON-MONOTONIC PARAMETER FUNCTION**

NmPF (Non-monotonic Parameter Function), a program written in SPlus, implements the procedure outlined above. The code for this function is listed in Appendix A. For the sake of this discussion, assume that the overall trend of the response curve is positive, i.e.,



$$r = \frac{\sum xp - n\bar{x}\bar{p}}{\sum x^2 - n\bar{x}^2} > 0,$$

where  $r$  is the slope coefficient of a simple linear regression of  $p$  on  $x$ . Starting with the first outcome value and stepping along the curve from smaller to larger values of the  $x$  variable, the program checks each pair of adjacent points, denoted  $p[i]$  and  $p[i+1]$ . The first point in each pair is compared with its immediate successor to determine whether or not  $p[i+1] - p[i] \geq 0$ . If not, the program uses the test statistic derived above to calculate the standardized  $z$ -value of  $Y = p[i+1] - p[i]$ . If the absolute value of the test statistic is greater than three, then the non-monotonic deviation from the overall trend of the response surface is construed as significant. That is, the magnitude of the non-monotonicity implies it is likely caused by some attribute of the model other than the random fluctuations of the stochastic parameters.

After calculating the  $z$ -values for the non-monotonic points on the response curve, the program sums the magnitudes of all the  $z$ -values and sets the sum equal to  $NmP$ .  $NmPF$  returns not only the  $NmP$  value, but also the number of non-monotonic jumps ( $J$ ) and the number of significant jumps ( $SJ$ ). Additionally,  $NmPF$  detects non-monotonic trends. A trend is defined as two or more successive non-monotonicities, i.e.,  $p[i+j] - p[i] < 0$ , where  $j = 1 \dots n$  and  $n > 1$ .  $NmPF$  returns the sum of all trends ( $NmT$ ), the number of non-monotonic trends ( $T$ ) and the number of significant non-monotonic trends ( $ST$ ) along the response curve. Keeping track of trends is necessary since a series of single, insignificant, non-monotonic jumps could result in a significant deviation from the overall trend of the response surface, but would not be picked up by the  $NmP$  parameter.

$NmPF$  continues to track each non-monotonic trend until there is either a single significant jump in the opposite direction, or the sum of two or more jumps in the opposite direction becomes significant. Once this last constraint is met,  $NmPF$  stops tracking the current non-monotonic trend and starts looking for the next non-monotonic deviation at the point where it stops tracking the current trend. Figure 5.3 illustrates the calculation of these parameters.

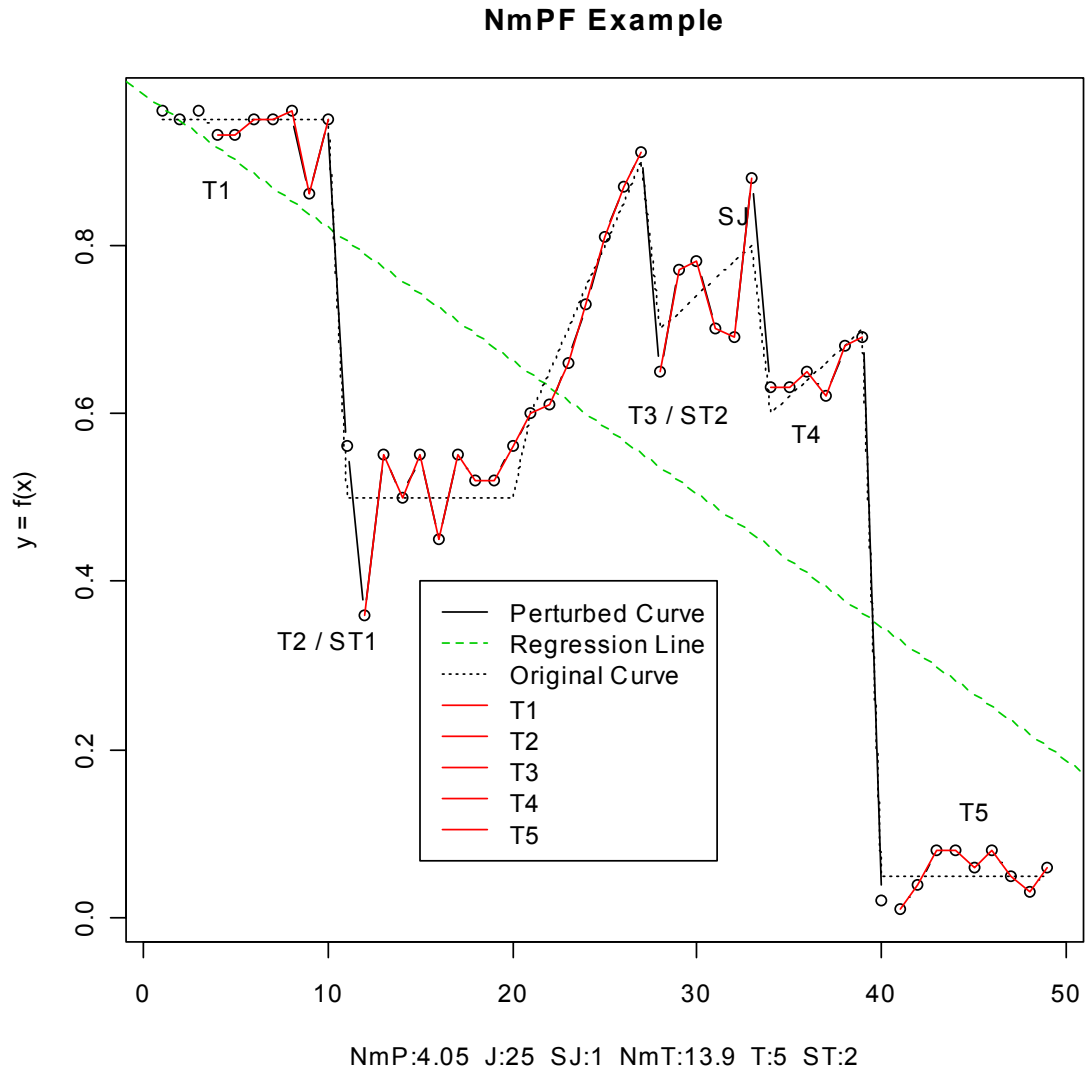


Figure 5.3. A Piecewise Linear, Non-monotonic Curve.

This graph illustrates the output of the NmPF parameter values for a known response curve. Just below the x-axis are the calculated NmPF parameter values for this curve. The sum of all non-monotonic jumps (NmP) is 4.05. There are 25 total jumps (J), only one of which is a significant jump (SJ). Hence, there are 24 insignificant jumps. The sum of all non-monotonic trends (NmT) is 13.9. There are also five trends (T), only two of which are significant (ST). Hence, there are three insignificant trends. Note that the NmT parameter value is more than three times as large as the NmP parameter value. This highlights the utility of the NmT parameter; it has picked up on both significant non-monotonic trends, whereas NmP missed the larger trend because it was made up of several insignificant jumps.

The original response curve in Figure 5.3 is clearly non-monotonic. Each point on the original curve is used as the probability of success in a Bernoulli trial. Each of the points on the graph, designated by small circles, is the mean of one thousand such Bernoulli trials. The overall trend of the perturbed points is indicated by the regression line that runs from the upper left corner to the lower right corner of the graph. The regression line indicates that the overall trend is decreasing. Thus, if  $p[i + 1] - p[i] > 0$ , then  $p[i + 1]$  is a non-monotonic deviation from the overall trend of the response curve.

Starting with the perturbed points in the upper left hand corner of the graph, observe that the third point from the left (small circle),  $p[3]$ , is non-monotonic. However, it is neither a significant non-monotonicity nor the start of a non-monotonic trend. Continuing along the response curve, from left to right, note that  $p[5]$  is the start of a non-monotonic trend; not only is  $p[6] - p[5] > 0$ , but so is  $p[7] - p[5] > 0$ . This upward trend continues up through  $p[8]$ .  $p[9]$  is a jump in the opposite direction. However, since  $p[9] - p[8]$  is not a significant jump and  $p[10] - p[4] > 0$ , NmPF continues tracking this trend up through  $p[11]$ .  $p[11] - p[10]$  is a significant jump in the opposite direction. Therefore, NmPF stops tracking the trend at  $p[11]$  and starts looking for the next non-monotonic deviation from the overall trend, starting with  $p[11]$ . In a similar manner, the remaining trends were identified and the NmPF values were calculated based on the total number of jumps and trends found along the response curve. Each of the five trends is highlighted in red and labeled from left to right by “Tx,” where x is the trend number. The two significant trends are labeled “ST1” and “ST2.” The last segment of the third non-monotonic trend is the only significant non-monotonicity (SJ) that occurred on this response surface. It is labeled “SJ.”

### C. TRANSFORMING THE Z-VALUES

As mentioned above, NmPF uses three standard deviations as the threshold for a significant non-monotonicity. However, three standard deviations is a sharp boundary, and numbers near this boundary, but slightly less in magnitude, could be real non-monotonicities and not just random fluctuations. Therefore, it does not seem appropriate to exclude them from consideration. For example, suppose that there are three successive points, each of which is a non-monotonic deviation from the overall trend of

the response surface, but each of them has an absolute z-value just less than 3.0, say 2.99. Then each of these points would not be included in the NmP value calculated by the program. Compare this with the case where there are three successive points with z-values slightly larger than 3.0, say 3.01. Since each of these latter points is larger than 3, they would all be included in the NmP value calculated by the program. The non-monotonicity in both cases is, for all practical purposes, identical. In the first case, if the non-monotonic points slightly less than 3.0 are the only non-monotonic points, then the NmP value would be zero. In the second case, if the non-monotonic points slightly larger than 3.0 were the only non-monotonic points, then the NmP value would be 9.03. Ideally, NmP should be influenced by all of these points, but not as much by those z-values whose magnitudes are less than 3. What is needed is a custom function,  $f$ , that equals 3 when  $z = 3$ , increases as the magnitude of  $z$  gets large and decreases to zero, quickly, as  $z$  approaches zero. So, define the following transformation of the  $z$  values:

$$f(d) = \begin{cases} 3e^{\lambda d} & d < 0 \\ 3 + d & d \geq 0 \end{cases},$$

where  $d = z - 3$ . Thus,  $f(d) \rightarrow 3$  as  $z \rightarrow 3$ ;  $f(d) \rightarrow 0$  as  $z \rightarrow 0$  and, finally,  $f(d)$  increases linearly when  $z > 3$ . This transformation makes the NmP values continuous. The additional parameter  $\lambda$ , allows the function to be more or less responsive to the threshold boundary. The graph in Figure 5.4 shows some representative values of  $\lambda$  and its effect on the behavior of  $f(d)$ .

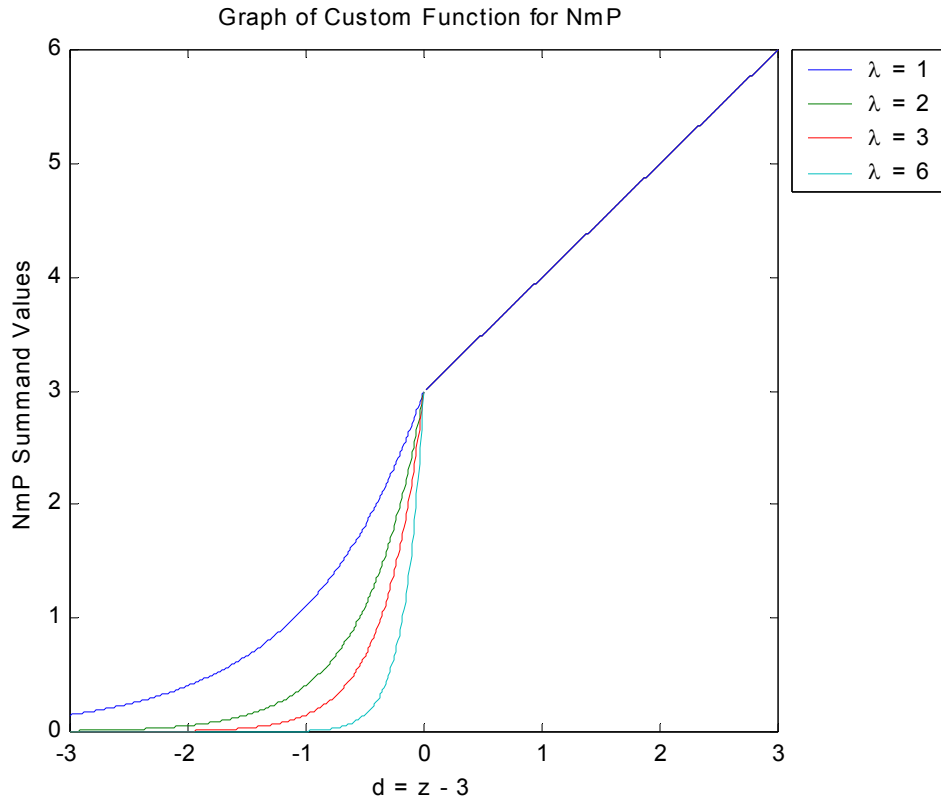


Figure 5.4. Z-value Transformation Function.

As  $\lambda$  Increases,  $f(d)$  approaches zero more quickly when  $d < 0$ . In the calculations that follow,  $\lambda = 6$  gives the most sensible behavior.

#### D. THE EMPIRICAL DISTRIBUTION OF THE NMPF VALUES

As noted at the end of section V.A above, the true distribution of the difference between points on the response surface is difficult to determine analytically. Hence, we determine the distribution of these points numerically. The NmPF parameters, also described above, measure various attributes of these numbers and so are parameters of the distribution of these points.

The actual code that calculates the NmPF parameter values is modified slightly so that it generates sensible output when non-monotonic jumps occur from one to zero, and vice versa. An adjustment factor of  $10^{-9}$  is added to the denominator of the z statistic to preclude division by zero. Thus, the actual formula used in the NmPF function is:

$$z = \left| \frac{(\hat{p}_2 - \hat{p}_1)\sqrt{n}}{\sqrt{\hat{p}_2(1 - \hat{p}_2) + \hat{p}_1(1 - \hat{p}_1) + 10^{-9}}} \right|,$$

where  $n$  is the number of replications. This adjustment factor has a negligible effect on the z statistic when data values are greater than 0.005 and less than 0.995. However, when data values are simultaneously close to zero and one, the z statistic becomes large. In those cases where a non-monotonic jump from one to zero, or vice versa, occurs, the z statistic is bounded above by

$$\frac{\sqrt{n}}{\sqrt{10^{-9}}} = \sqrt{n10^9}.$$

When the deterministic model is run,  $n = 1$ , and the value of a single non-monotonic jump is bounded above by  $\sqrt{10^9} = 31,622.78$ . When the stochastic model is run,  $n = 1000$ , and the value of a single non-monotonic jump is bounded above by  $\sqrt{1000 \cdot 10^9} = 10^6$ . Figure 5.5 displays the z statistic surface, computed without the adjustment factor (left), and with the adjustment factor (right). The  $\hat{p}_1$  and  $\hat{p}_2$  values used to generate these graphs are contained in the interval  $[0.005, 0.995]$ . The graph clearly shows that the adjustment factor has no effect on the z statistic for response surface values bounded by 0.005 and 0.995.

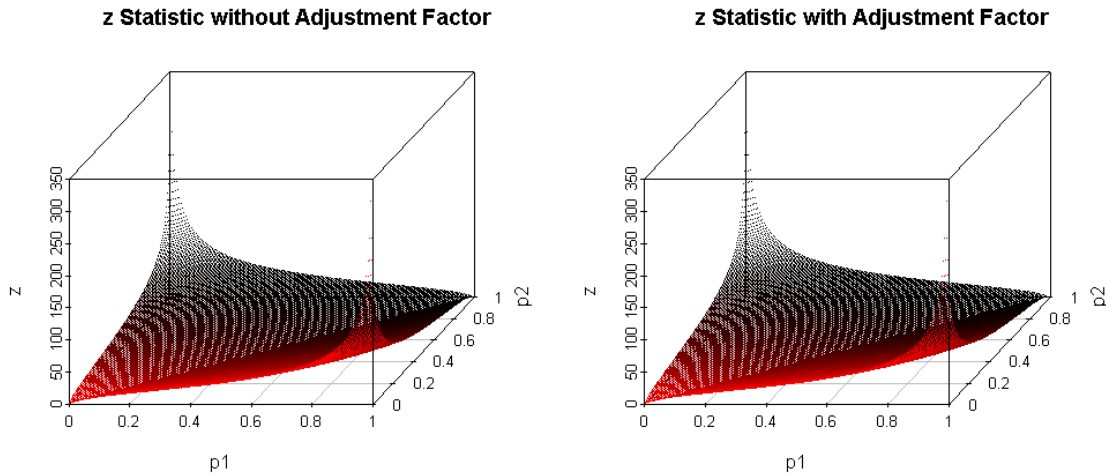


Figure 5.5. Adjustment Factor for z Statistic Does Not Affect Results

To determine the distribution of the NmPF values, Monte Carlo simulations are run in which NmPF is tested on several known response curves. Two monotonic curves and three non-monotonic curves are randomly perturbed, thus generating a stochastic response curve—i.e., a curve whose values are bounded by 0 and 1. Since the true underlying curves are known, the parameters generated by the NmPF can be tested to see if they accurately reflect the behavior of these curves. The random binomial generator in SPlus is used as the stochastic parameter of the Monte Carlo simulation.

The first monotonic curve tested is the horizontal line  $y = 0.5$ . This curve is used since the stochastic perturbations generated along this line have the greatest variance. The variance of any other probability response curve generated over a monotonic surface in this manner must be smaller. The results of this simulation will provide a “worst case” with which we can compare unknown surfaces.

Two vectors are created, an  $x$  vector and a  $y$  vector. The  $x$  vector contains the same number of elements as a typical slice of the Dewar model’s response surface. The values of  $x$  are the sequence of numbers from ten to 2000 in increments of ten. Thus, there are 200 elements in the  $x$  vector. The second vector, the  $y$  vector, contains the same number of elements as  $x$ . For the first simulation, each of the  $y$  values is equal to 0.5. Each of these 200  $y$  values is used as the ‘probability of success’ argument in the

binomial random generator function in SPlus—i.e., the true probability of a Red Win at each  $x$  value. 1000 Bernoulli random variates are generated for each of the elements of the  $y$  vector. The mean of each of these one thousand trials is then calculated and saved in a third vector called  $yp$ . After each of the elements in  $y$  is randomly perturbed in this manner, we draw the response curve by plotting  $x$  vs  $yp$ . The first two graphs in the top row of Figure 5.6 illustrate the input and output of this process. The third graph in the top row shows the simple regression line that represents the overall trend of the data—i.e., whether the general trend of the response curve is increasing or decreasing. Once created, the new vector, containing the means of 1000 Bernoulli trials, is passed as an argument to NmPF. NmPF calculates the following parameters for the response curve:

- a. NmP—The sum of the magnitudes of all non-monotonicities.
- b. Jumps (J)—The number of non-monotonicities.
- c. Significant jumps (SJ) —The number of Jumps whose magnitude is greater than 3.
- d. NmT—The sum of magnitudes of all non-monotonic trends.
- e. Trends (T)—The number of non-monotonic trends.
- f. Significant Trends (ST)—The number of Trends whose magnitudes are greater than 3.

This entire process is repeated one thousand times. For each replication, NmPF calculates the parameter values listed above and then generates a histogram for each of them. The last six graphs in Figure 5.6 display the histograms, along with means and standard deviations for each NmPF parameter.

We repeat this Monte Carlo simulation for a monotonic decreasing surface. The results are graphically displayed in Figure 5.7. Note that the NmPF parameter values are smaller than the “worst case” scenario shown in Figure 5.6. Figures 5.7 – 5.9, the results of the simulations of the three known non-monotonic curves, show that the NmPF parameters behave as expected when used to measure response surfaces whose true underlying surface is non-monotonic. The results of these five simulations indicate that the NmPF parameters may be useful in measuring the non-monotonicity of a response curve. However, further testing is necessary.



## Non-Monotonic Parameter Distribution

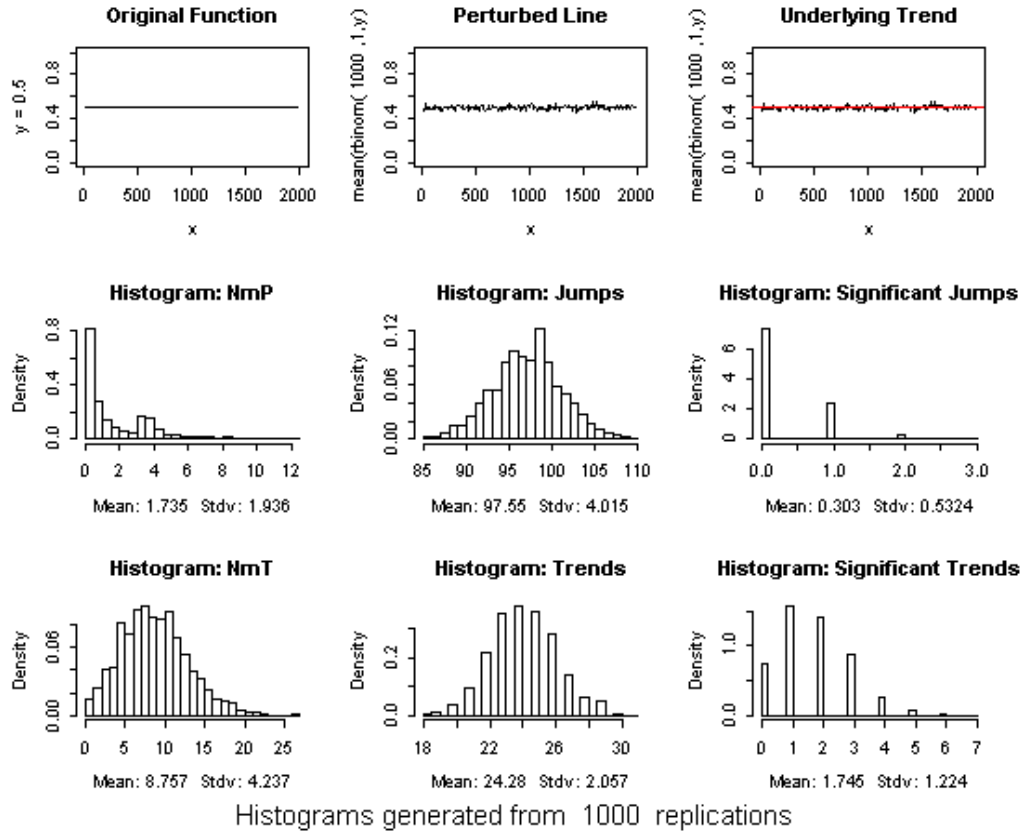


Figure 5.6. A Constant, Monotonic Function.

Stochastic perturbation creates non-monotonicity. How often are these non-monotonicities significant? Any successive point that is higher than its predecessor is non-monotonic. The simple linear regression slope coefficient for this response surface should be zero. However, due to stochastic perturbation it is not. Since the regression coefficient is not zero, the NmPF parameters only count non-monotonic deviations from the overall trend of the response curve. In this particular case, that equates to roughly half of the points on the response curve (random perturbation of the points on  $p = 0.5$  distributes roughly half the perturbed points above 0.5 and the other half below 0.5). According to the simulation results, when the true surface is monotonic, random fluctuations greater than three standard deviations happen, on average, about 0.3 times out of a thousand. Likewise, significant trends happen, on average, about 1.8 times out of a thousand. Since the variance of the Bernoulli distribution is greatest when the probability of success is 0.5, the NmPF values from this simulation equate to a “worst case” scenario, given that the underlying surface is truly monotonic. Compare this with Figures 5.6 – 5.9.

## Non-Monotonic Parameter Distribution

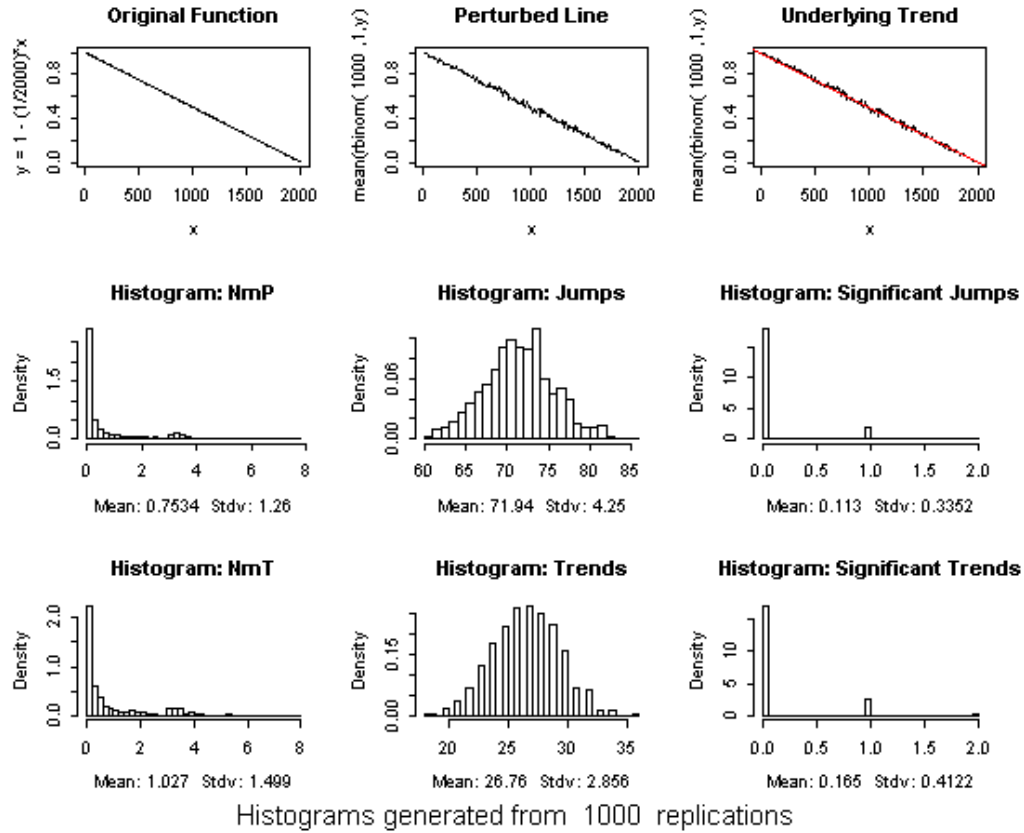


Figure 5.7. A Decreasing, Monotonic Function.

The regression line in the top right corner graph indicates the overall trend of the data is decreasing. Thus, moving along the response curve from left to right, any successive point that is higher than its immediate predecessor is non-monotonic. The graphs in the second and third rows indicate that, among the non-monotonicities, random fluctuations greater than three standard deviations occur, on average, about 0.11 times out of a thousand. Note that the decrease in the values of NmP, S, SJ, NmT and ST from the corresponding values in Figure 5.6. Since the variance of the binomial is smaller, on average, for this surface than for the surface in Figure 5.6—the variance is smaller, on average, because the surface is decreasing—the fluctuations are also smaller, hence the smaller NmPF parameter values. Note that the overall number of jumps remains fairly constant. The points on the response curve are normally distributed; hence, the random fluctuations are almost as likely to be above the true surface as below. In each simulation, there are the same number of original points (200), so we expect that roughly half of the stochastic points will be non-monotonic. This trend is fairly consistent throughout Figures 5.5 – 5.9

## Non-Monotonic Parameter Distribution

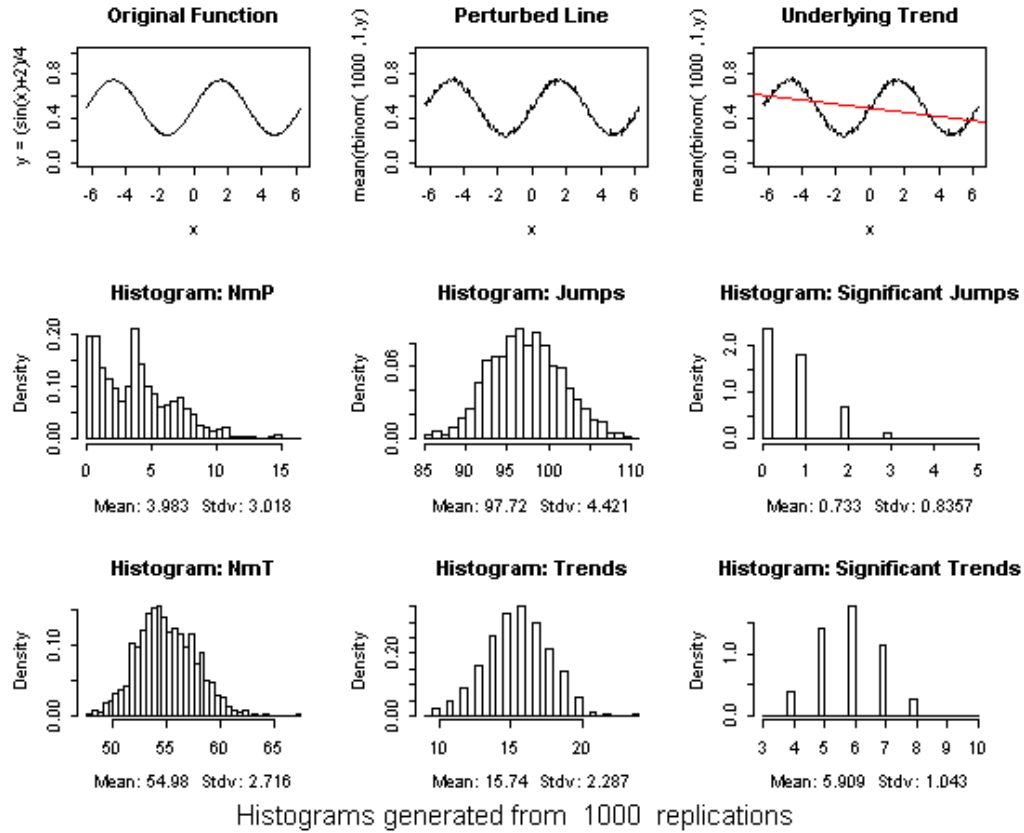
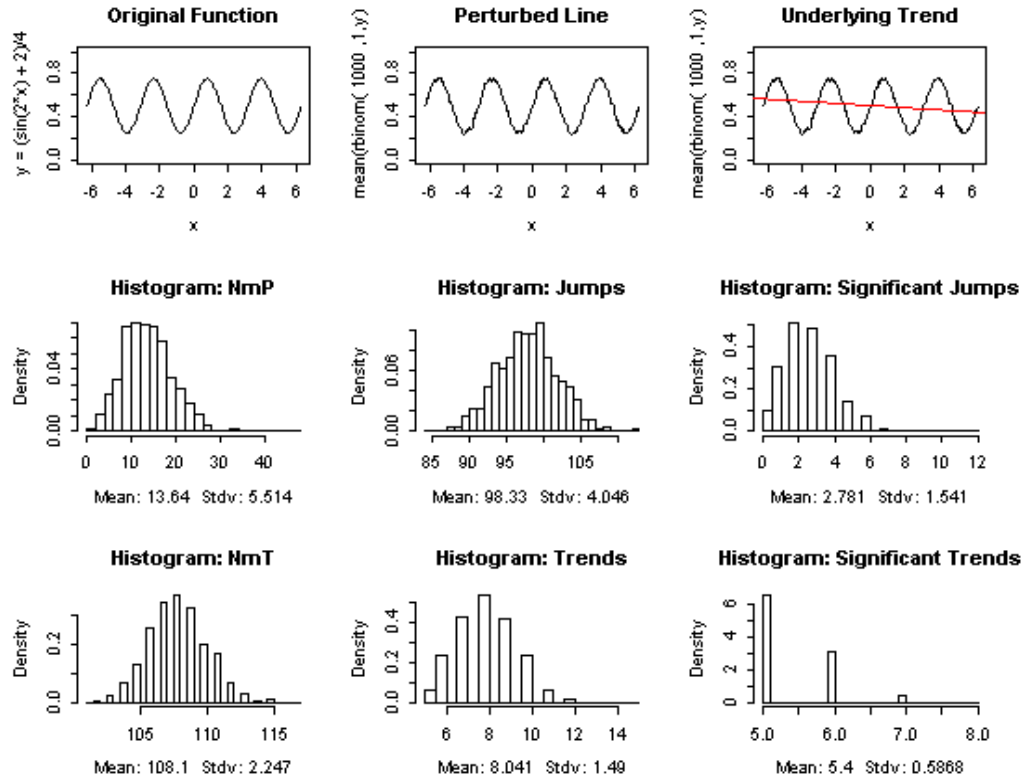


Figure 5.8. An Extremely Non-monotonic Function.

Large, non-monotonic trends are obtained by the following equation:  $y = \frac{\sin x + 2}{4}$ .

Here, underlying "truth" is non-monotonic; hence, the NmPF parameter values should be larger than when underlying "truth" is monotonic, and they are. Note that, when true non-monotonic trends in the response curve are present, NmT is a better indicator than NmP of non-monotonic behavior. If the underlying trend is fairly smooth, and the points are fairly dense, there will be few significant single jumps, and so NmP will be small. However, since the true underlying curve has long, pronounced non-monotonic trends, the NmT value identifies these as significant. The multi-modal histogram of NmP reflects the distribution of the significant jumps. Based on the values in Figure 5.6, we would fail to classify this response curve as significantly non-monotonic using only the NmP parameter. Thus, the NmT value is also needed.

## Non-Monotonic Parameter Distribution



Histograms generated from 1000 replications

Figure 5.9. What Happens If the Frequency Is Doubled?

If the non-monotonicity is doubled, we would expect some of the NmPF parameter values to at least double, and indeed they do; in fact, NmP and SJ more than triple in value compared to Figure 5.8. However, the Trends value is reduced by fifty percent, while ST remains almost constant. This seemingly non-intuitive behavior is caused by the stretching of the response curve. Since each response curve has the same number of points, and the length of this curve has doubled, the point density along this response curve has decreased. Consequently, there are fewer chances for non-monotonic trends, unless the curve is truly non-monotonic. Again, the number of overall non-monotonic jumps remains approximately constant. According to the “worst case” numbers in Figure 5.6, we would classify the curve in Figure 5.9 as significantly non-monotonic.

## Non-Monotonic Parameter Distribution

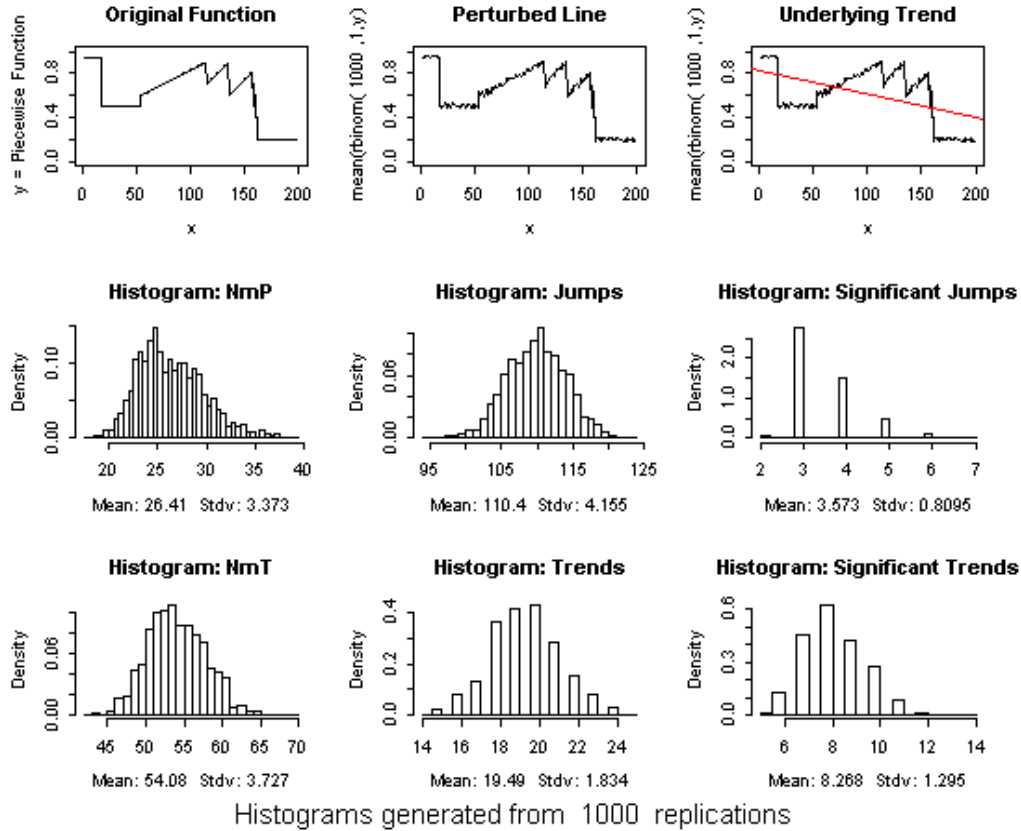


Figure 5.10. A Non-monotonic, Piecewise Function.

Since the actual curve underlying the response curve is non-monotonic, the number of significant jumps and trends should be higher than in Figure 5.6, and they are. Here, there are not the same long, pronounced non-monotonic trends as in Figures 5.7 and 5.8. So, in this sense, the non-monotonicity is more tame than in those figures. However, this irregular curve has more Jumps and Significant Jumps. Hence, its NmP value is almost twice as large as the NmP value in Figure 5.9. Based on the “worst case” numbers from Figure 5.6, we would classify the response curve in Figure 5.10 as significantly non-monotonic.

In all five Monte Carlo simulations, the number of points along the response curve remains constant; the only changes are in the shape and inclination of the curve. The results, displayed graphically in Figures 5.5 – 5.9, show that the NmPF parameters behave in an intuitive manner. Additionally, the results of the first simulation provide limiting values, or upper bounds, which can be used to classify the non-monotonicity of unknown response curves, based on their NmPF parameter values. Table 5.1 lists these worst-case limiting values. The number of jumps, significant jumps, trends and significant trends are discrete rather than continuous quantities. The use of discrete values seems more appropriate. Therefore, Table 5.1 also lists quantiles for the right tails of the NmPF parameter distributions. Note that the quantile values closely match the three standard deviation values. NmPF values greater than  $|\mu \pm 3\sigma|$  we define to be significant; this corresponds, roughly, to the 99.5<sup>th</sup> quantile of the empirical values.

	$\mu$	$\sigma$	$\mu+3\sigma$
<b>NmP</b>	1.7310	1.9650	7.6260
<b>Jumps</b>	97.5300	4.1090	109.8570
<b>SJumps</b>	0.2948	0.5371	1.9061
<b>NmT</b>	8.7930	4.2250	21.4680
<b>Trends</b>	24.3300	2.0580	30.5040
<b>STrends</b>	1.7520	1.2430	5.4810

	<b>Quantiles</b>						
	<b>0.5%</b>	<b>1.0%</b>	<b>10.0%</b>	<b>80.0%</b>	<b>90.0%</b>	<b>99.0%</b>	<b>99.5%</b>
<b>NmP</b>	0.02	0.02	0.09	3.46	4.29	7.93	9.02
<b>Jumps</b>	87.00	88.00	92.00	101.00	103.00	107.00	108.00
<b>SJumps</b>	0.00	0.00	0.00	1.00	1.00	2.00	2.00
<b>NmT</b>	0.65	0.95	3.68	12.30	14.50	19.90	20.90
<b>Trends</b>	19.00	20.00	22.00	26.00	27.00	29.00	30.00
<b>STrends</b>	0.00	0.00	0.00	3.00	3.00	5.00	6.00

Table 5.1. NmPF Parameter Means, Standard Deviations and Quantiles For the Worst Case Monotonic Curve.

From Table 5.1, we conclude that if the underlying trend of the response curve is truly monotonic, then we do not expect to see NmPF parameter values greater than the estimated values listed in the 99.5<sup>th</sup> quantile column. A comment is necessary here. We have set the significance level at, roughly, three standard deviations (the 99.5<sup>th</sup> quantile) to increase the probability that curves classified as non-monotonic truly are non-monotonic. By doing so, we have increased the probability of a type II error—that is,

failing to classify a curve as non-monotonic when it really is. Our primary interest is in identifying practically significant non-monotonicities. A lower significance level runs the risk of identifying statistically significant, but practically insignificant, non-monotonicities. Therefore, we feel that the increased risk of a type II error is justified.

Up to this point, the NmPF has been tested only on curves. However, the output of each fractional factorial experimental run will be a two-dimensional surface. We need to know limiting NmPF parameter values for surfaces. As described above, the argument to NmPF is the pair of vectors representing the projection of a slice of the response surface onto the x-z plane, where x is the Blue parameter variable, y is the Red parameter variable, and z is the vector containing the values of the given response surface slice. In the discussion above, the Blue and Red parameter variables used were initial force levels. Red was fixed at 500, and the corresponding Blue initial force levels and response surface values were fed to the NmPF. The NmPF then calculates the parameter values for that particular slice of the response surface. The method for calculating the NmPF parameter values for a surface is exactly the same, except that all of the response surface slices corresponding to unique values of y are input to the NmPF one at a time and a cumulative total for each of the parameter values is computed. The cumulative totals are then returned as NmPF parameter values for the surface.

The same procedure used to develop upper bounds for the NmPF parameter values for curves is used to develop limiting values for the NmPF parameter values for surfaces. However, only one Monte Carlo simulation is run. The single run is for the worst-case monotonic surface—i.e., the horizontal plane  $z = 0.5$ . Figure 5.11 shows the distributions of the NmPF parameter values for this worst-case surface. Table 5.2 is a listing of the means, standard deviations and right tail quantiles for the NmPF parameter values for the worst-case surface. Note that the estimated means of the surface values are equal to the curve value estimates, scaled by a factor of 349 ( $\pm 0.005$  due to random variation). Of particular interest is the fact that the NmPF parameter values for the worst case monotonic response surface indicate the tendency towards normality that is guaranteed by the Central Limit Theorem. The 99.5% quantiles in Table 5.2 match the three standard deviation values also listed in Table 5.2. Therefore, we claim that the

NmPF parameter values for surfaces are normally distributed as the number of points on the response surface grows large. Certainly, 69451 points—the number of points on a response surface generated by the stochastic version of the Dewar model—are enough to satisfy the normality assumption.

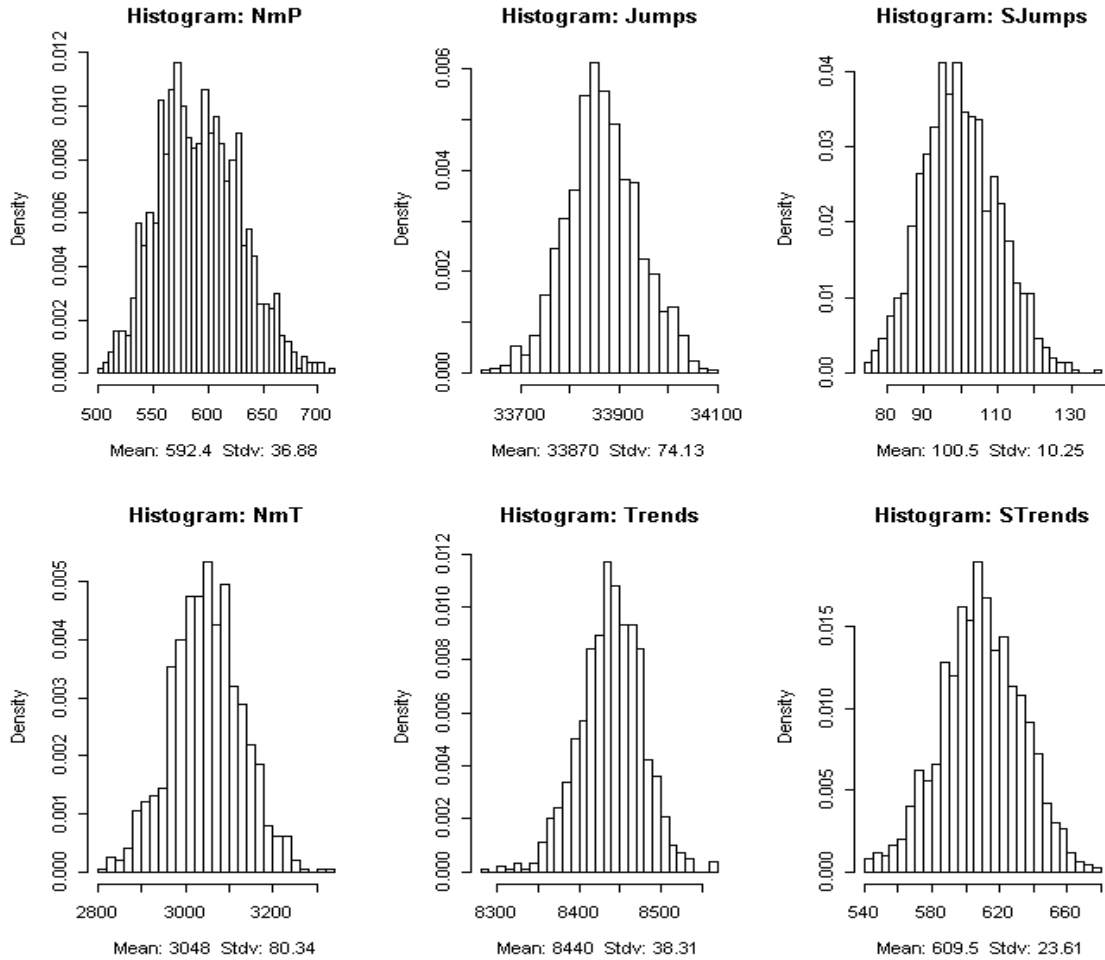


Figure 5.11. Histograms of NmPF Parameter Values For a Surface.

These histograms were generated by a Monte Carlo simulation, using the NmPF to calculate parameter values for the stochastically perturbed surface,  $z = 0.5$ . The simulation was run for 1000 iterations. Each response surface generated during the fractional factorial experiment has the same number of points: 69,451. These points are plotted over a 199 x 349 grid. It takes NmPF approximately five minutes to calculate the NmPF parameter values for one of these surfaces. The Monte Carlo simulation, used to generate Table 5.2, took approximately 5000 minutes or 3.47 days to complete. See Table 5.2 for a listing of the right tail quantile estimates.



	$\mu$	$\sigma$	$\mu+3\sigma$
<b>NmP</b>	592.4	36.88	703.04
<b>Jumps</b>	33870.0	74.13	34092.39
<b>SJumps</b>	100.5	10.25	131.25
<b>NmT</b>	3048.0	80.34	3289.02
<b>Trends</b>	8440.0	38.31	8554.93
<b>STrends</b>	609.5	23.61	680.33

**Surface Quantiles**

	90.0%	99.0%	99.5%
<b>NmP</b>	641	680.5	694.6
<b>Jumps</b>	33970	34040.0	34040.0
<b>SJumps</b>	114	125.0	127.0
<b>NmT</b>	3152	3231.0	3247.0
<b>Trends</b>	8485	8529.0	8536.0
<b>STrends</b>	640	662.0	666.0

Table 5.2. NmPF Parameter Means, Standard Deviations and Quantiles For the Worst Case Monotonic Surface.

**E. ANALYSIS OF NMPF VALUES WITH RESPECT TO  $\lambda$ , POINT DENSITY, NUMBER OF REPLICATIONS AND TYPE OF RESPONSE SURFACE.**

A full factorial design experiment is conducted to confirm the observations mentioned above and to refine our understanding of the behavior of the NmPF values as the following factors change:  $\lambda$  (z-transformation tuning parameter); the number of points in a given interval (point density); the number of replications used to generate each response curve point; or the shape of the response curve. The full factorial design experiment uses the following factors and levels:

	<b>Levels</b>				
<b>Factors</b>	1	2	3	4	5
<b>Point Density</b>	200	400			
$\lambda$	1	6			
<b>Replications</b>	100	500			
<b>Shape of Response Curve</b>	f1	f2	f3	f4	f5

Figure 5.12 shows the functions that determine the shape of the response curve used in the full factorial design. Knowing the true shape underlying the response curve allows us

to draw conclusions about the behavior of the NmPF values obtained by stochastically perturbing the curves. For each run, the following algorithm is used:

a.  $y = f(x)$  is generated, where  $x$  is a vector of evenly spaced values between zero and one. The number of elements in  $x$ , and  $f(x)$  are as prescribed by the experimental design.

b.  $yp = \text{mean}(\text{rbinom}(1000,1,y))$  is generated for each curve. This perturbs the original curve, generating the stochastic response curve.

c.  $\text{NmPF}(x, yp, n, \lambda, \text{thresh})$  is called. The arguments are:  $n$ , the number of replications;  $\lambda$ , the tuning parameter for the  $z$ -value transformation described earlier;  $x$  and  $yp$ , the independent and dependent values of the response curve;  $\text{thresh}$ , the significance level in terms of standard deviations. The thresh value is set at three for the reasons discussed in section VI.D. The NmPF parameter values are returned and stored.

d. Step (c) is repeated 1000 times for each design point. The means of the NmPF parameter values for each design point are then returned.

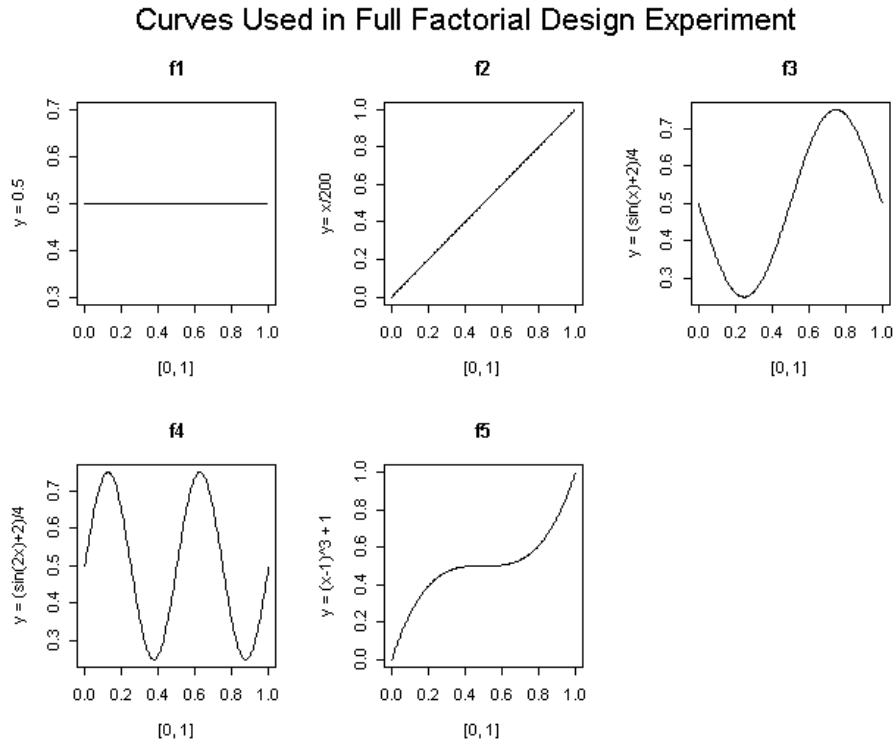


Figure 5.12. Known Response Curves Used in Full Factorial Design Experiment.

## F. RESULTS OF FULL FACTORIAL DESIGN EXPERIMENT

The experimental results, the ANOVA analysis, and calculated effects are listed in detail in Appendix B. The following conclusions are drawn from the experiment's results and the subsequent analysis:

### 1. ANOVA Results for NmP

The values in Table 5.3 indicate that NmP is extremely sensitive to the value of  $\lambda$ . This makes sense. If  $\lambda$  is small, say  $\lambda = 1$ , and the response curve has many insignificant jumps, then the sum of all the insignificant jumps can grow quite large as the number of insignificant jumps increases. If  $\lambda$  is large, say  $\lambda = 6$ , then the effect of all the insignificant jumps is dramatically reduced. Thus, the experimental results confirm our intuition and verify the expected behavior of  $\lambda$ . NmP is also fairly sensitive to the number of points. This is no surprise. More points imply more opportunities for non-monotonicities. More non-monotonicities imply a larger NmP value. The interaction between  $\lambda$  and Density is also straightforward and intuitive. Low  $\lambda$  and high point density results in large NmP values; conversely, high  $\lambda$  and low point density results in low NmP values, exactly what is expected.

	Df	Sum of Sq	Mean Sq	F Value	Pr (F)
fNames	4	223.819	55.955	45.678	0.0000000
<b>Density</b>	<b>1</b>	<b>1023.471</b>	<b>1023.471</b>	<b>835.503</b>	<b>0.0000000</b>
<b>Lambda</b>	<b>1</b>	<b>7316.173</b>	<b>7316.173</b>	<b>5972.501</b>	<b>0.0000000</b>
SampleSize	1	0.117	0.117	0.095	0.7613839
fNames:Density	4	3.845	0.961	0.785	0.5506660
fNames:Lambda	4	128.536	32.134	26.232	0.0000004
fNames:SampleSize	4	41.022	10.255	8.372	0.0006365
<b>Density:Lambda</b>	<b>1</b>	<b>891.953</b>	<b>891.953</b>	<b>728.139</b>	<b>0.0000000</b>
Density:SampleSize	1	2.205	2.205	1.800	0.1973829
Lambda:SampleSize	1	0.894	0.894	0.730	0.4048281
Residuals	17	20.825	1.225		

Table 5.3. ANOVA Summary For the NmP Parameter.

### 2. ANOVA Results for Jumps (J)

The values in Table 5.4 indicate that the J parameter is extremely sensitive to the number of points along the response curve. This happens for the same reason as mentioned above for the NmP parameter. This extreme sensitivity should present no problem as long as the stochastic variables used in the model are fairly symmetric. Asymmetrically distributed stochastic variables might cause the value of J to be biased

either high or low, depending on the asymmetry of the distribution. The stochastic version of the Dewar model uses only symmetric random variables.

	Df	Sum of Sq	Mean Sq	F Value	Pr (F)
<b>fNames</b>	<b>4</b>	<b>1590.31</b>	<b>397.58</b>	<b>2379.0</b>	<b>0.0000000</b>
<b>Density</b>	<b>1</b>	<b>22127.62</b>	<b>22127.62</b>	<b>132404.2</b>	<b>0.0000000</b>
Lambda	1	0.12	0.12	0.7	0.4150663
SampleSize	1	4.46	4.46	26.7	0.0000773
fNames:Density	4	7.59	1.90	11.4	0.0001134
fNames:Lambda	4	0.22	0.06	0.3	0.8507183
<b>fNames:SampleSize</b>	<b>4</b>	<b>148.11</b>	<b>37.03</b>	<b>221.6</b>	<b>0.0000000</b>
Density:Lambda	1	0.07	0.07	0.4	0.5392657
Density:SampleSize	1	6.38	6.38	38.2	0.0000101
Lambda:SampleSize	1	0.05	0.05	0.3	0.5900005
Residuals	17	2.84	0.17		

Table 5.4. ANOVA Summary For the J Parameter.

The Jump parameter is also fairly sensitive to the shape of the curve. Here is a list of the calculated effects for the five curves:

Effect of Curve on J:		
1	5.03900	$y = 0.5$
2	-8.87975	$y = 1 - (1/2000) * x$
3	4.97275	$y = (\sin(x) + 2) / 4$
4	5.31525	$y = (\sin(2x) + 2) / 4$
5	-6.44725	$y = (x-1)^3 + 1$

The first curve is the “worst case,”  $y = 0.5$ . A positive value for the effect indicates that this horizontal, constant curve increased the value of J, on average, by five. The number 2 curve, a monotonic decreasing curve, reduced the value of J, on average, by almost nine. The 3<sup>rd</sup> and 4<sup>th</sup> curves, the  $\sin(x)$  curves, increased the value of Jumps, on average, by about five. The 5<sup>th</sup> curve, the monotonic increasing curve  $y = (x - 1)^3 + 1$ , reduced the value of J, on average, by about six. These values verify that the horizontal line  $y = 0.5$  represents the worst-case scenario when the true, underlying curve is monotonic. They also emphasize the intuitive behavior of the J parameter.

### 3. ANOVA Results for Significant Jumps (SJ)

Even though three of the factors in Table 5.5 have a statistically significant effect on the behavior of SJ, the F-values for the SJ parameter are orders of magnitude smaller than those for either J or NmP. The conclusion to draw from this is that SJ is relatively unaffected by the factors in the experiment. This is a good thing. SJ should be affected only by the magnitude of non-monotonic jumps. Clearly, the point density and the shape

of the curve will affect the number of jumps (more jumps imply a greater chance for significant jumps) and the magnitude of the jumps (a rapidly decreasing surface will have more significant jumps). This explains the significance of point density and curve shape on SJ. However, compared to the NmP and J parameters, the SJ parameter is relatively unaffected by the factors used in this experiment.

	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
<b>fNames</b>	<b>4</b>	<b>0.50899</b>	<b>0.1272475</b>	<b>57.48625</b>	<b>0.000000</b>
<b>Density</b>	<b>1</b>	<b>0.10816</b>	<b>0.1081600</b>	<b>48.86314</b>	<b>0.0000022</b>
Lambda	1	0.00225	0.0022500	1.01648	0.3274921
SampleSize	1	0.00729	0.0072900	3.29338	0.0872449
fNames:Density	4	0.06534	0.0163350	7.37962	0.0012296
fNames:Lambda	4	0.00305	0.0007625	0.34447	0.8441677
<b>fNames:SampleSize</b>	<b>4</b>	<b>0.21991</b>	<b>0.0549775</b>	<b>24.83703</b>	<b>0.0000007</b>
Density:Lambda	1	0.00064	0.0006400	0.28913	0.5977474
Density:SampleSize	1	0.03364	0.0336400	15.19745	0.0011557
Lambda:SampleSize	1	0.00009	0.0000900	0.04066	0.8425906
Residuals	17	0.03763	0.0022135		

Table 5.5. ANOVA Summary For the SJ Parameter.

#### 4. ANOVA Results for NmT

NmT is the one NmPF value that is sensitive to all four factors: shape of the curve, point density,  $\lambda$ , and the number of replications used to generate each response curve point. However,  $\lambda$  dominates the other three. Because NmT is the sum of all non-monotonic trends, it will be affected by  $\lambda$  more or less depending on the number of insignificant trends; more insignificant trends imply larger NmT value. This explains the dominance of  $\lambda$  in Table 5.6.

	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
<b>fNames</b>	<b>4</b>	<b>3961.816</b>	<b>990.454</b>	<b>745.305</b>	<b>0.0000000</b>
<b>Density</b>	<b>1</b>	<b>829.489</b>	<b>829.489</b>	<b>624.181</b>	<b>0.0000000</b>
<b>Lambda</b>	<b>1</b>	<b>1644.121</b>	<b>1644.121</b>	<b>1237.182</b>	<b>0.0000000</b>
SampleSize	1	98.820	98.820	74.361	0.0000001
fNames:Density	4	8.388	2.097	1.578	0.2254740
fNames:Lambda	4	109.118	27.279	20.527	0.0000025
fNames:SampleSize	4	550.328	137.582	103.529	0.0000000
Density:Lambda	1	305.761	305.761	230.082	0.0000000
Density:SampleSize	1	5.022	5.022	3.779	0.0686411
Lambda:SampleSize	1	22.945	22.945	17.266	0.0006627
Residuals	17	22.592	1.329		

Table 5.6. ANOVA Summary For the NmT Parameter.

Next to  $\lambda$ , as expected, the greatest effect on NmT is caused by the shape of the curve. As the curve becomes more non-monotonic, the value of NmT increases. The following are the calculated effects of the five curves used in the experiment.

#### Effect of Response Curve Shape on NmT:

1	-3.143360	$y = 0.5$
2	-10.172060	$y = 1 - (1/2000)*x$
3	5.365765	$y = (\sin(x) + 2)/4$
4	16.653265	$y = (\sin(2x) + 2)/4$
5	-8.703610	$y = (x-1)^3 + 1$

The negative numbers in the above list indicate that the value of NmT decreased, on average, by the amount shown. Positive numbers indicate an average increase. Note that all of the monotonic functions (1, 2, 5) have negative values, while all of the non-monotonic functions (3, 4) have positive values. This confirms that the value of NmT increases as the non-monotonicity of a curve increases. Exactly the behavior required.

### 5. ANOVA Results for Trends (T)

As Table 5.7 shows, T is affected almost solely by point density. The relative magnitude of Density's F-value implies that the number of points along the response curve has the greatest effect on the value of the T parameter. In this, T behaves in almost the same manner as J.

	Df	Sum of Sq	Mean Sq	F Value	Pr (F)
fNames	4	124.859	31.215	60.650	0.0000000
<b>Density</b>	<b>1</b>	<b>2004.773</b>	<b>2004.773</b>	<b>3895.254</b>	<b>0.0000000</b>
Lambda	1	0.038	0.038	0.075	0.7879211
SampleSize	1	40.280	40.280	78.265	0.0000001
fNames:Density	4	20.686	5.172	10.048	0.0002317
fNames:Lambda	4	0.075	0.019	0.036	0.9972132
fNames:SampleSize	4	22.076	5.519	10.723	0.0001591
Density:Lambda	1	0.011	0.011	0.021	0.8860570
Density:SampleSize	1	4.706	4.706	9.144	0.0076552
Lambda:SampleSize	1	0.042	0.042	0.082	0.7779455
Residuals	17	8.749	0.515		

Table 5.7. ANOVA Summary For the T Parameter.

One notable difference is that the effect of the shape of the curve on the value of the T parameter is exactly opposite to that of all the other parameters. This can be seen in the following table, where the calculated effects of the curves on Trends is highlighted in red. The calculated effects, of the curves, for all other parameters are positive.

NmP:	Jumps:	SJumps:	NmT:	Trends:	STrends:
1 0.9709625	1 5.03900	1 -0.01825	1 -3.143360	1 0.20025	1 -0.75850
2 -3.1095375	2 -8.87975	2 -0.12825	2 -10.172060	2 2.49900	2 -1.88600
<b>3 1.4835000</b>	<b>3 4.97275</b>	<b>3 0.03925</b>	<b>3 5.365765</b>	<b>3 -0.05475</b>	<b>3 1.61525</b>
<b>4 3.0615875</b>	<b>4 5.31525</b>	<b>4 0.19425</b>	<b>4 16.653265</b>	<b>4 -3.02850</b>	<b>4 2.60900</b>
5 -2.4065125	5 -6.44725	5 -0.08700	5 -8.703610	5 0.38400	5 -1.579

The reason for this behavior is not immediately obvious. As the shape of the surface becomes more non-monotonic, the number of insignificant trends tends to decrease, especially along point-sparse curves, due to the way the T parameter is

generated (for a complete description, refer to section V.B and Figure 5.3.). When the response curve is fairly ‘monotonic,’ there tend to be a lot of insignificant trends. As the response curve grows more non-monotonic, these trends tend to become significant and longer; thus, there are fewer insignificant trends and fewer trends overall. For smooth, continuous, non-monotonic response curves, the value of the T parameter and the ST parameter approach a common value (refer to Figures 5.7 and 5.8). For non-smooth, non-monotonic response curves this is not the case—i.e., the T parameter value continues to increase (see Figure 5.10).

## 6. ANOVA Results for Significant Trends (ST)

As expected, ST is primarily affected by the shape of the response curve (see Table 5.8). The number of points along the response curve has an effect on the number of non-monotonicities and, hence, the number of trends. Therefore, ST is also affected by the point density. ST is immune to the effect of  $\lambda$ , as it should be. By design,  $\lambda$  affects only insignificant trends.

	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
<b>fNames</b>	<b>4</b>	<b>128.3507</b>	<b>32.08768</b>	<b>275.7747</b>	<b>0.0000000</b>
<b>Density</b>	<b>1</b>	<b>13.7358</b>	<b>13.73584</b>	<b>118.0514</b>	<b>0.0000000</b>
Lambda	1	0.0000	0.00000	0.0000	1.0000000
SampleSize	1	0.1188	0.11881	1.0211	0.3264251
fNames:Density	4	6.0321	1.50803	12.9607	0.0000509
fNames:Lambda	4	0.0129	0.00323	0.0278	0.9983528
fNames:SampleSize	4	4.8321	1.20802	10.3822	0.0001920
Density:Lambda	1	0.0130	0.01296	0.1114	0.7426558
Density:SampleSize	1	0.2402	0.24025	2.0648	0.1688842
Lambda:SampleSize	1	0.0036	0.00361	0.0310	0.8622640
Residuals	17	1.9780	0.11635		

Table 5.8. ANOVA Summary For the ST Parameter.

Next, the NmPF function is applied to actual data sets generated by the stochastic version of the Dewar Modal. After that, we make some concluding remarks regarding the NmPF and appropriate settings for its use.

## G. APPLYING THE NMPF TO ACTUAL DATA

Figures 5.12 – 5.16 illustrate the behavior of the NmPF values on actual data obtained from experimental runs of the stochastic version of the Dewar model. Each figure displays two rows of graphs. Each graph is a projection of the response surface onto either the Blue-z plane or the Red-z plane. The top row in each figure shows

projections of the response surface where Blue values are held constant. The bottom row shows projections of the response surface where Red values are held constant. Figures 5.12 and 5.15 show NmPF values generated with  $\lambda = 1$ . Figures 5.14 and 5.16 show NmPF values generated with  $\lambda = 6$ .

Compare the NmPF values in Figure 5.13, with  $\lambda = 1$ , to the NmPF values in Figure 5.15, with  $\lambda = 6$ . In the upper left corner of Figure 5.13, (Blue = 500),  $NmP = 60$ . Compare this with the NmP value from the same graph in Figure 5.15, where  $NmP = 4.08$ . There are 129 jumps in this slice of the response surface, only one of which is significant. The large NmP value in Figure 5.13 is due to the large number of insignificant jumps, coupled with the fact that  $\lambda = 1$ . In this case, the large NmP value is caused by the random noise of the stochastic variables used to generate the response curve. Setting  $\lambda = 6$ , as in Figure 5.15, filters out this noise. Consequently, the NmP value in Figure 5.15 is dramatically smaller. The NmT value behaves in a similar manner, although the NmT parameter value in Figure 5.15 decreases only by a factor of about two. This change is not quite as dramatic as the order of magnitude change exhibited by the NmP value as  $\lambda$  changes from one to six. In both cases, increasing  $\lambda$  has the desired effect: it reduces the effect of the insignificant non-monotonicities.

Now, observe the graph in the lower left corner of Figure 5.13. According to the NmPF parameters, there are seven jumps, four of which are significant, and two trends, both of which are significant. The same graph in Figure 5.15 shows a decrease in the value of NmP as  $\lambda$  changes from one in Figure 5.13 to six in Figure 5.15. However, the NmT value remains constant, as it should.

Next, the middle graph, on the bottom row of Figures 5.12 and 5.14, displays 21 jumps, one of which is significant, and three trends, two of which are significant. Also notice that the non-monotonic jumps occur around 0.4 – 0.6, in the range where the variance is greatest. The non-monotonic trends in the graph are very pronounced, visually.

According to the 99.5% quantiles in Table 5.1, we would classify all the graphs in Figure 5.13, but only the two leftmost graphs in Figure 5.15, as significantly non-



monotonic. However, using the 90% quantile from Table 5.1, the lower middle graph in Figure 5.15 also would be classified as significantly non-monotonic.

The bottom right graph is troublesome. The visually pronounced dip in this graph seems to make it classifiable as significantly non-monotonic. But the NmPF parameters imply that it is not. There is one significant non-monotonic trend; however, the quantiles in Table 5.1 indicate that this significant non-monotonic trend could have happened by chance more than 10% of the time. See the enlarged view of this curve in Figure 5.14. It is important to note that one small, but statistically significant jump (or trend) in an otherwise ‘monotonic’ curve may not be caught. This is an example of the type II error that can occur with our approach.

The NmPF parameter values displayed in Figures 5.15 and 5.16 behave in a similar fashion. However, these graphs demonstrate typical numbers seen when the response curve is extremely non-monotonic. As expected, the NmPF parameter values reflect this extreme non-monotonicity.

## Non-Monotonic Parameter Values

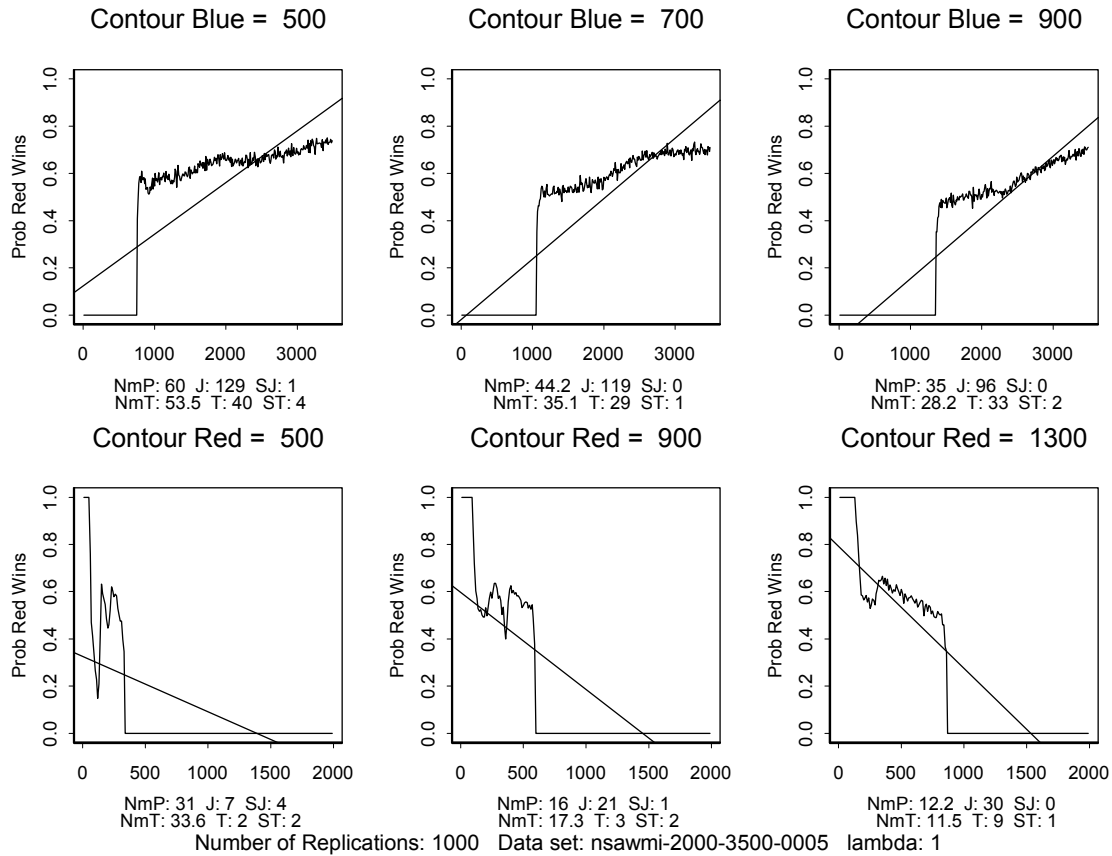


Figure 5.13. Is the Response Curve Non-Monotonic or Not?

In this set of graphs,  $\lambda = 1$ , where  $\lambda$  is a tuning parameter for the NmPF values that makes them more or less sensitive to the  $3\sigma$  boundary. Note the behavior of the response surface and how the NmPF parameter values reflect this behavior. Compare these graphs with those in Figure 5.14.

A Closer Look at The Non-Monotonicity in Figure 6.12

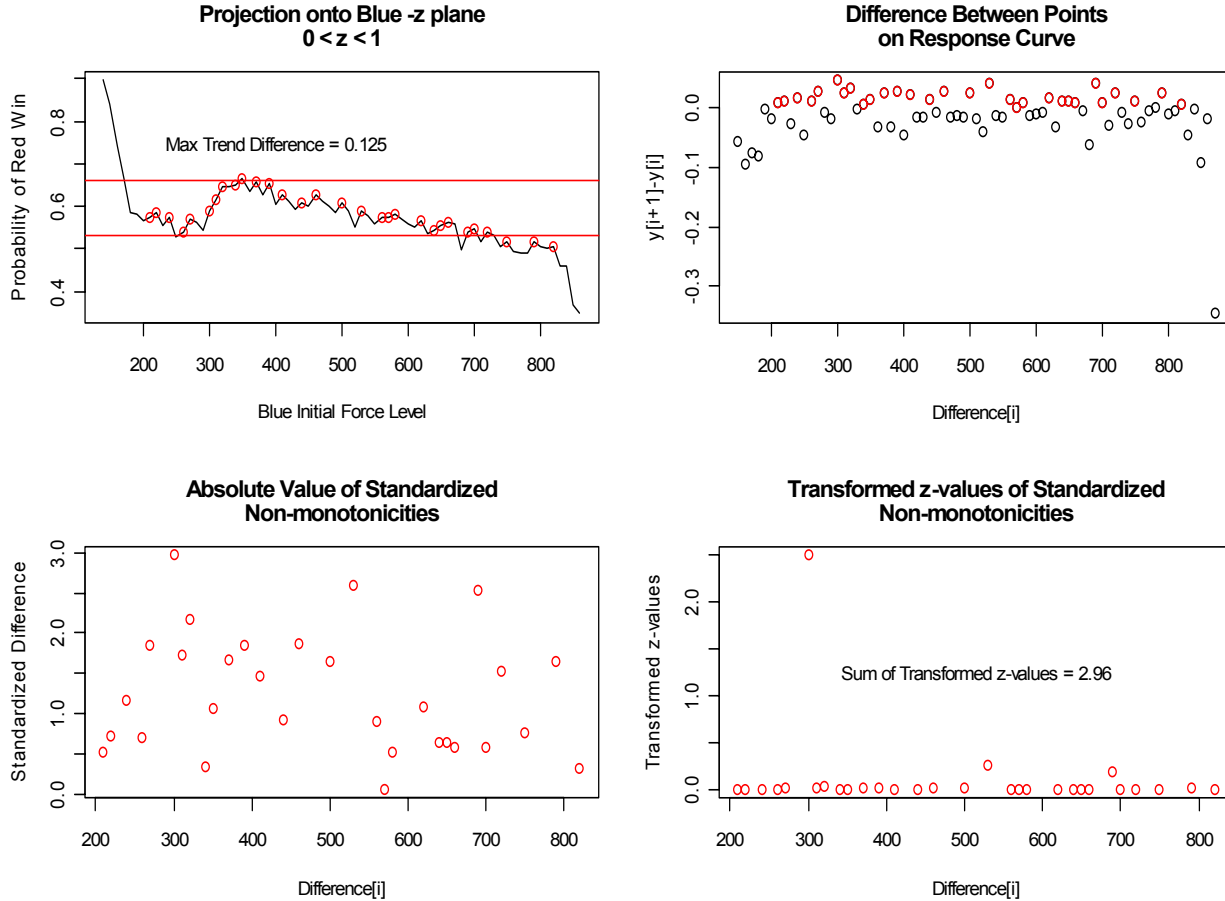


Figure 5.14. A Close-up View of Figure 5.13.

This Figure shows an enlarged view of the non-monotonic points of the bottom right graph in Figure 5.13. The non-monotonic trend is readily apparent. The difference between the lowest and highest points on this trend is 0.125. The z-value for this difference is 6.21. Each of the points plotted in red is a non-monotonic point.

## Non-Monotonic Parameter Values

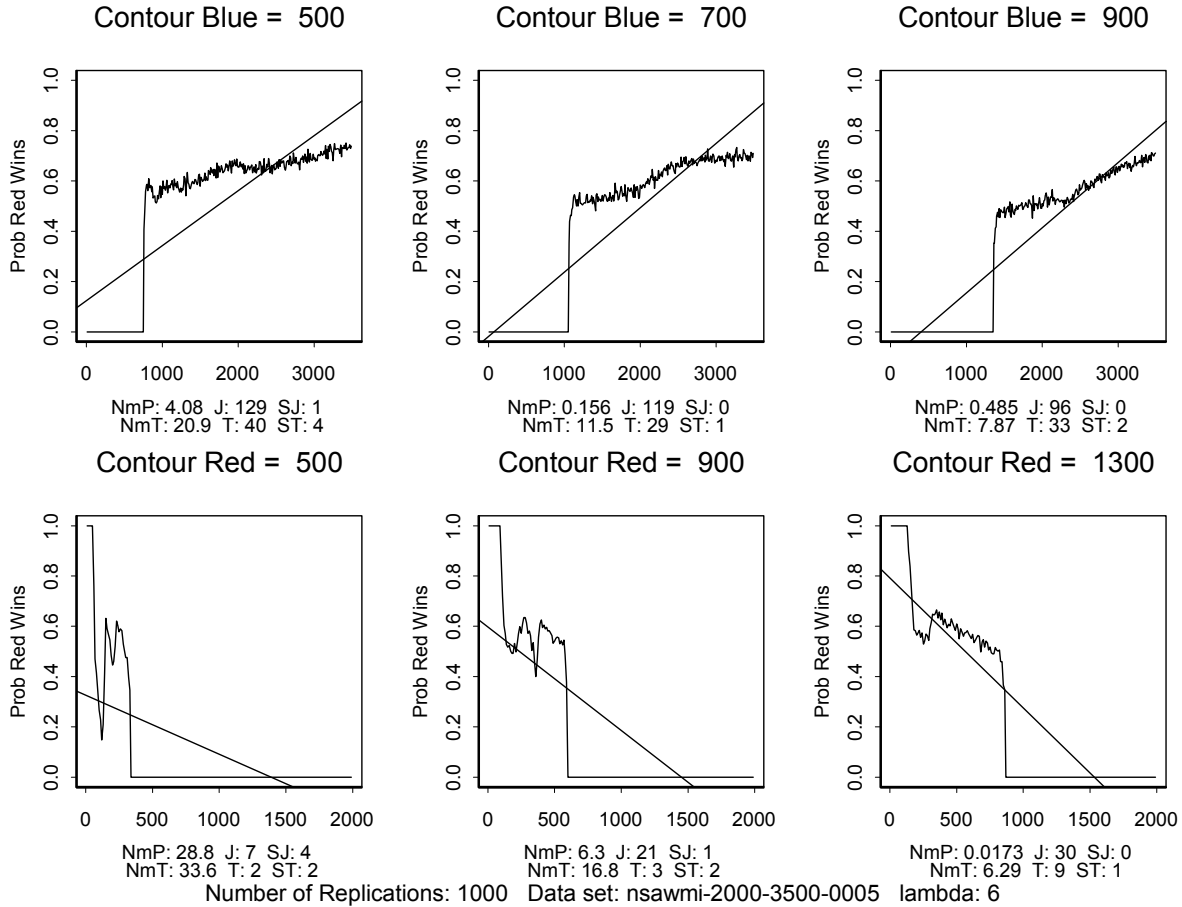


Figure 5.15.  $\lambda$  Reduces the Effect of Random Noise on the NmPF Values.

In this set of graphs,  $\lambda = 6$ . Note the behavior of the response surface and how the NmPF parameter values reflect this behavior. Compare these graphs with those in Figure 5.13. According to the worst-case numbers from Figure 5.6, only the graph in the lower left corner of Figure 5.15 would be classified as significantly non-monotonic.

## Non-Monotonic Parameter Values

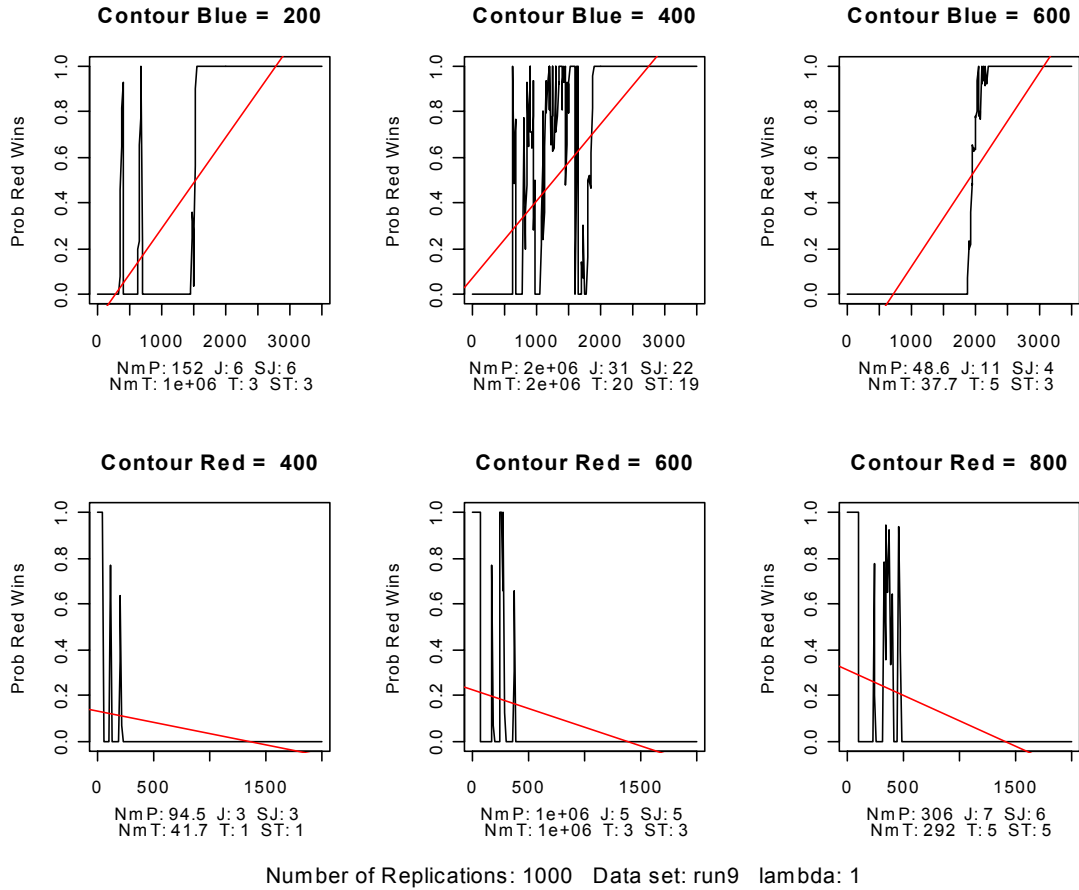


Figure 5.16. NmPF Parameter Values and Extreme Non-Monotonic Behavior.

In this set of graphs,  $\lambda = 1$ . Note the extremely non-monotonic behavior of these graphs and how the NmPF parameter values reflect this behavior. Compare these graphs with those in Figure 5.17. Note that, except for the graphs in the upper right corner, the NmP and NmT values do not change. This is caused by the fact that  $J = SJ$  and  $T = ST$  in seven out of the 12 cases. Except for the graphs in the upper right corner, the difference in the NmP and NmT values as  $\lambda$  changes from one to six in the other five cases is lost due to the magnitude of the NmP and NmT parameter values.

## Non-Monotonic Parameter Values

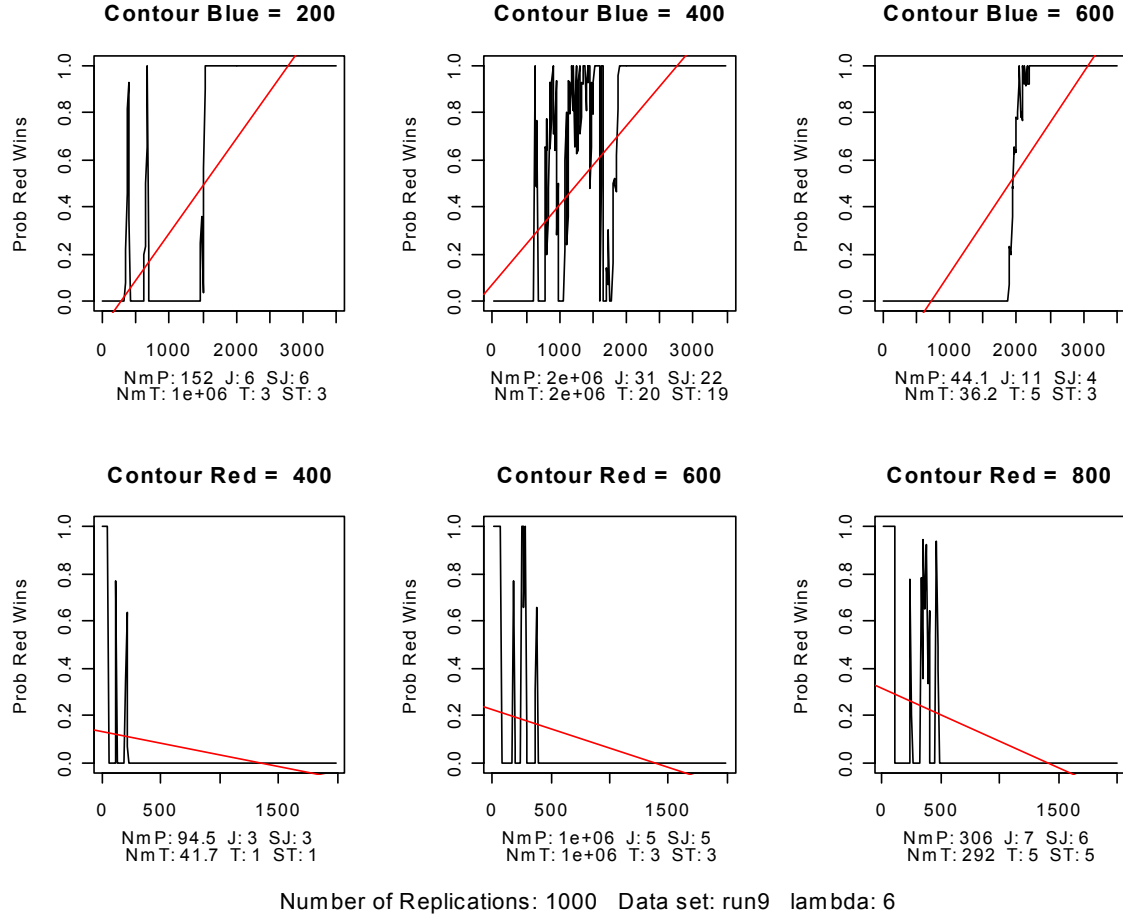


Figure 5.17. The Effect of  $\lambda$  on NmPF Parameter Values and Extreme Non-monotonic Behavior.

In this set of graphs,  $\lambda = 6$ . Note the extreme non-monotonic behavior of these graphs and how the NmPF values reflect this behavior. Since almost all of the non-monotonicities are significant, the NmP and NmT values change very little, if at all, when  $\lambda$  is changed from one to six. This is another sign that the NmPF function is working properly. Compare the NmPF parameter values in these graphs with those in Figures 5.12 – 5.15.

## H. CONCLUDING REMARKS REGARDING THE NMPF

Based on all of the above—the discussion, the experiment, and the application of the NmPF to actual data—we are confident that the NmPF parameter values will effectively classify the response surfaces generated by the stochastic version of the Dewar model. To summarize the salient points developed thus far:

1. The points on the response surfaces generated by the stochastic version of the Dewar model are independent and normally distributed; so is any single difference between any pair of points on the surface. However, multiple successive differences are not independently distributed. To measure the non-monotonicity of these surfaces, multiple successive comparisons are necessary. The analytical derivation of the distribution of multiple successive comparisons being too difficult, the distribution and error rates are obtained numerically by Monte Carlo Simulation and the development of a Non-monotonic Parameter Function (NmPF).

2. The NmPF is evaluated using a full factorial experimental design. Each of the NmPF parameters behaves consistently and in an intuitive manner. The main findings of the factorial experiment are:

- a. As the number of points across the response surface increases, the NmPF parameter values become approximately normally distributed. This assumption is further supported by the fact that the NmPF parameters, Jumps, SJumps, Trends and STrends are all discrete counting variables and are naturally inclined towards a Poisson distribution. However, as the mean of a Poisson distribution grows large, the Poisson distribution may also be approximated by the Normal distribution. This is exactly the behavior the NmPF parameter values exhibit for response surfaces.

- b. The effect of insignificant non-monotonicities on the NmPF parameter values, due only to the randomness added to the model by the stochastic parameters can be dramatically reduced by  $\lambda$ , the NmPF tuning parameter. This parameter makes the NmPF parameters more sensitive to significant non-monotonicities and is set to six for future use.

3. A standardized, non-monotonic deviation from the overall trend of the response surface, whose absolute magnitude is greater than three, is considered “significant.” The use of  $3\sigma$  rather than  $2\sigma$  increases the possibility of a type II error. A lower significance level, while reducing the probability of a type II error, increases the chance of identifying statistically significant, but practically insignificant, non-monotonicities. The increased risk of a type II error is justified in order to increase the probability that significant non-monotonicities are practically significant.

4. Two tables are generated which provide upper bounds for the right tails of the NmPF parameter distributions. The first table contains means, standard deviations and quantiles for response curves; the second contains the same information for response surfaces. The means and standard deviations of the two tables are identical to within random variation and a scale factor of 349. Table 5.2 provides a quantifiable basis for discriminating between surfaces that contain insignificant non-monotonicities and those that contain significant non-monotonicities.

The next section covers the results of the fractional factorial experiment and conclusions drawn from it with regard to the effect of stochastic perturbations on the non-monotonicity exhibited by the Dewar model.



THIS PAGE INTENTIONALLY LEFT BLANK

## VI. FRACTIONAL FACTORIAL EXPERIMENT RESULTS

This chapter analyzes the results of the fractional factorial experiment and calculates the effects of making the parameters stochastic. These results are tested by applying the information obtained from the analysis to the model and making a final series of runs to compare the analytical and empirical results.

### A. ANALYSIS

According to the design developed in Chapter IV, 64 design points are run through the model. This requires  $64 \times 1000 \times 69451$ , or approximately 4.5 billion runs. The NmPF is run on each of the 64 response surfaces generated by the 64 experimental design points. The result is a 64 by six matrix, containing the six NmPF parameter values for each of the 64 response surfaces. Table C1 in Appendix C is a listing of the design matrix and the results of the experiment.

#### 1. Correlation Among the Responses and the Method of Analysis

According to the discussion in Chapter V, we expect the response variables to be correlated. Multivariate analysis of variance (MANOVA) is a statistical method for analyzing multiple responses. “If the responses are independent, then it is sensible to just perform univariate analysis. However, if the responses are correlated, then MANOVA can be more informative than the univariate analyses as well as less repetitive” (*S-Plus 2000*, 1999). The only difference between the multivariate and the univariate analysis is that the variance of the calculated effects will be larger using MANOVA than when using ANOVA. MANOVA requires that the response variables satisfy the assumptions of independence and normality. As discussed in Chapter V, the NmPF parameter values are approximately normal when the NmPF is applied to the response surfaces generated by the stochastic version of the Dewar model. However, as we also noted in Chapter V, the NmPF parameters Jumps, SJumps, Trends and STrends are counting variables. Hence, they are likely distributed as Poisson random variates. Thus, a ‘square root’ transformation of the response variables may be required in order to ensure that the residuals are more normal. Additionally, the NmPF parameter values calculated for one response surface are independent of the NmPF parameter values calculated for any other

response surface. Thus, the response variables are approximately independent and identically distributed normal random variates. This follows from the fact that, when the mean is large, a Poisson random variable has approximately a normal distribution (Devore 1995, p. 237).

## 2. Graphical Diagnostics: Confirming Correlation

Figure 6.1 is a pairs plot of the response variable matrix. By inspection, it is obvious there is positive correlation between Jumps and Trends, as expected. Also obvious by inspection is the tight clustering of points that appears in each of the paired plots. A quick check confirms that this clustering is caused by the four orders of magnitude difference in the NmP and NmT parameter values as the factor level of Attrition Coefficients (AC) changes from ‘not stochastic’ to ‘stochastic.’ These clusters correspond to runs 33 through 64, the runs where the AC factor level is ‘stochastic.’

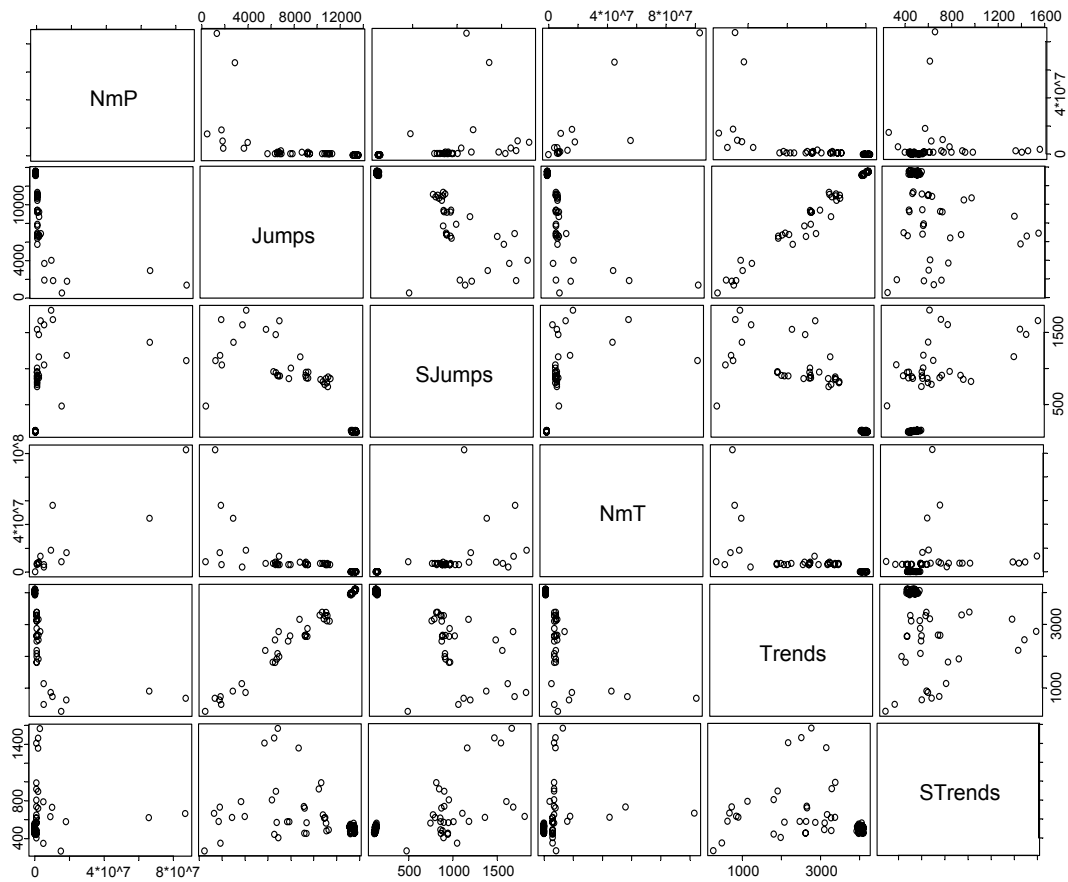


Figure 6.1. Pairs Plot: Response Variables From the Fractional Factorial Experiment.

	NmP	Jumps	SJumps	NmT	Trends	STrends
NmP	1.000	-0.541	0.331	0.884	-0.528	0.041
Jumps		1.000	-0.841	-0.629	0.989	-0.313
SJumps			1.000	0.510	-0.816	0.600
NmT				1.000	-0.608	0.181
Trends					1.000	-0.207
STrends						1.000

Table 6.1. Response Variable Correlation Matrix.

The correlation matrix in Table 6.1 confirms the visual impression of correlation between Jumps and Trends. It also shows that there is additional correlation in the data that is not so obvious in Figure 6.1. The positive and negative correlations in Table 6.1 whose magnitudes are greater than 0.5 are highlighted in blue and red, respectively.

### 3. Describing the Landscape of the Stochastic Response Surface

The left and right graphs in Figure 6.2 are representative of the response surfaces from runs 1–32 and 33–64, respectively. The black and white regions in each graph represent the initial force levels that result in a Blue and Red win, respectively. The colored region between the black and white regions indicates, according to the color bar, an increasing probability of a Red win. Clearly, the intermediate color region—where the probability that Red wins is greater than zero and less than one—is much larger in the right graph than in the left graph. The nearly homogeneous coloring of this region in the right graph indicates a nearly constant probability, except at the edges, adjacent to the black and white regions.

The NmPF parameter values displayed below Figure 6.2 correspond to the response surfaces of the left and right graphs, respectively. There is strong negative correlation between Jumps and Trends and the other four NmPF parameter values. So, while the number of Jumps and Trends increase, the values of NmP, NmT, SJumps and STrends all decrease, indicating a reduction in the non-monotonicity of the response surface. This general trend corresponds to the change in the AC factor from ‘not stochastic’ (left graph) to ‘stochastic’ (right graph).

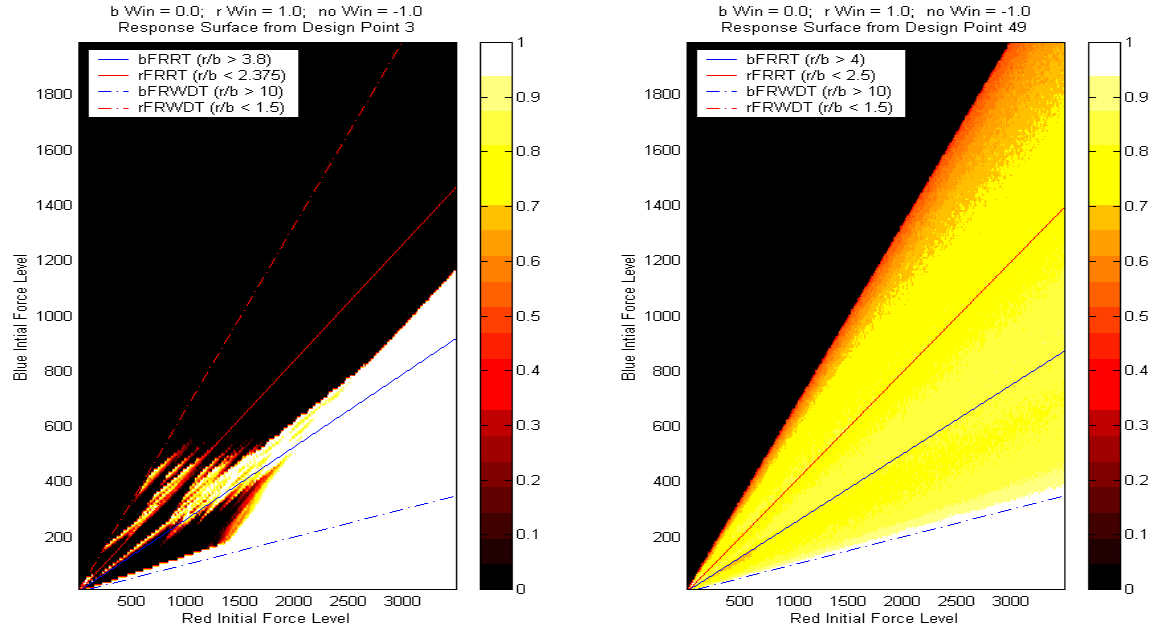


Figure 6.2. Explaining Correlation.

	NmP	Jumps	SJumps	NmT	Trends	STrends
<b>Design Point 3</b>	10028930	1822	1681	56021760	733	730
<b>Design Point 49</b>	808	13487	140	2852	4059	560

#### 4. Explaining the Negative Correlation

The cause of the negative correlation, while not immediately obvious, is easily explained. In Figure 6.2, observe that the region between the force ratio withdrawal thresholds (FRWDT in the graph legends) contains all of the non-monotonicity in both stochastic response surfaces. In this region of the graph on the left, the non-monotonicity is extreme—almost all of the Jumps and Trends are significant. However, in the right graph, the non-monotonic region is very mild—with relatively small SJumps and STrends parameter values. The parameter values Jumps and Trends have both increased by an order of magnitude, while the SJumps parameter value has decreased by an order of magnitude and STrends by almost 30 percent. All of the insignificant jumps and trends in the right graph are just the noise associated with the randomness added by the stochastic perturbations. In the right graph, the number of significant jumps and trends (SJumps, STrends) has decreased and, consequently, so have the values of NmP and NmT. All of the above comments are relative between the two graphs. However, in Chapter V, we generated the empirical distribution of the NmPF parameters. These

empirical distributions can be used to classify the stochastic surfaces as ‘monotonic’ or not.

## 5. Classifying a Stochastic Response Surface

The necessary quantiles can be found in Table 5.2. Clearly, the left graph in Figure 6.2 would be classified as extremely non-monotonic. The graph on the right would be classified as non-monotonic, but only barely so, and only because the NmP and Jumps values are a little too large. Additionally, as the cross-sections in the graphs in Figure 6.3 show, the trend of the response surface of design point 49 is dramatically smoother compared to the response surface of design point 3. These graphs demonstrate, again, the negative correlation between the NmPF parameter values, explained above.

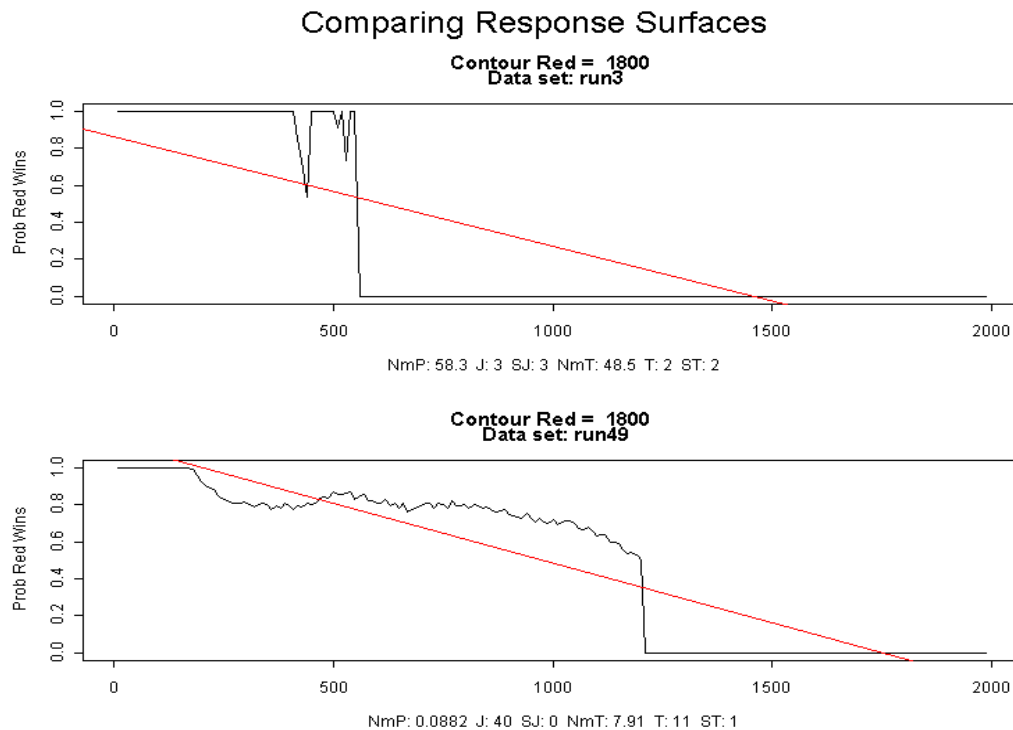


Figure 6.3. Explaining Negative Correlation Among NmPF Parameter Values.

The number of Jumps and Trends has increased from run 3 to run 49, but the number of SJumps and STrends has decreased. NmP and NmT are designed to be quite sensitive to SJumps and STrends, respectively. Therefore, there is strong positive correlation between NmP and SJumps, as well as between NmT and STrends. An extreme non-monotonic stochastic surface tends to have many significant jumps and trends, whereas a ‘monotonic’ stochastic surface tends to have few significant jumps and trends and a much larger number of insignificant jumps and trends. Also, according to Table 5.1, the cross-section of the response surface from run 49 is ‘monotonic’, but the one from run 3 is not.

## 6. Fitting the Model

Due to the strong correlation (positive and negative) among the NmPF parameter values, MANOVA seems to be the most suitable method for the analysis of the fractional factorial experiment. However, we omit the NmP and NmT response variables from the model, as the SJumps and STrends parameters provide the same information. Since we are interested in not only the main effects, but also first order interactions, a linear model with interactions is fitted. As previously mentioned, transforming the response variables Jumps, SJumps, Trends and STrends by taking their square roots may improve the normal behavior of the residuals in the fitted model. However, two models are fitted for comparison. The first model is fitted using the untransformed response variables, and the second is fitted using the square roots of the response variables. The diagnostic plots in Figure 6.4 indicate that the square root transformation is a more appropriate model. In Table 6.2, we have listed only those factors and interactions whose p-value is less than 0.01. The most obvious observation is that the attrition coefficients (AC) are the most significant parameters.

	Df	Approx F	num Df	den Df	Pr(>F)
FRRT	1	58.01	4	23	1.105e-11
PIFRT	1	31.72	4	23	4.675e-09
RBS	1	22.70	4	23	1.051e-07
AC	1	566.00	4	23	2.2e-16
PIFWDT	1	23.80	4	23	6.860e-08
RBA	1	5.74	4	23	0.002347
FRRT:AC	1	34.59	4	23	2.021e-09
PIFRT:AC	1	31.38	4	23	5.182e-09
PIFRT:PIFWDT	1	4.47	4	23	0.008114
RBD:FRWDT	1	6.08	4	23	0.001721
RBS:AC	1	21.64	4	23	1.612e-07
RBS:FRWDT	1	22.93	4	23	9.592e-08
Residuals	26				
RSE:					
Jumps		SJumps	Trends	STrends	
6.39		3.20	3.71	3.31	

Table 6.2. Summary of MANOVA Linear Model with Interactions

Table 6.2 summarizes the Linear model with interactions, fitted with the square root of the response variables. Factors and two-factor interactions have a significant effect on NmPF parameter values. Attrition coefficients are the most significant factors in the model. All decision thresholds and reinforcement block size, have a significant effect, either as single factors or as part of some interaction. The number of reinforcement blocks (RBA) also has a significant effect, but it is much less pronounced.

MANOVA Model: Linear with Interactions  
Comparing Un-transformed Residuals vs Square Root of Residuals

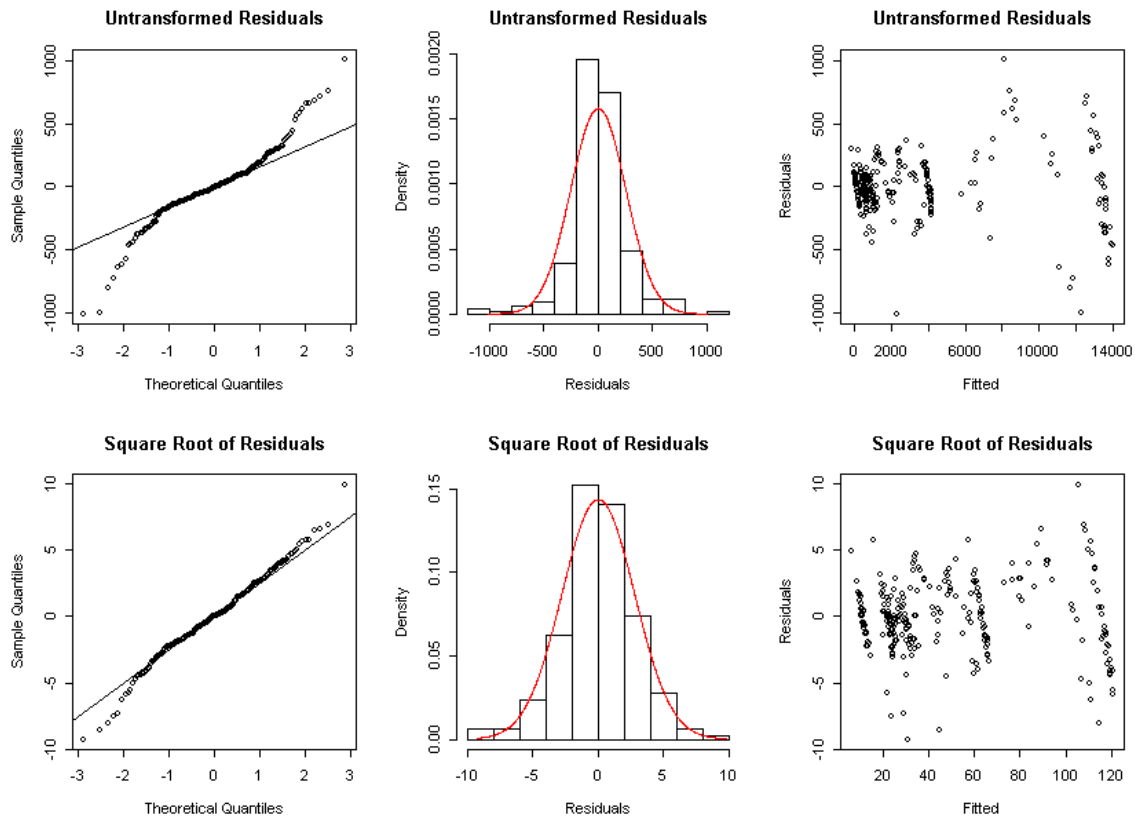


Figure 6.4. Checking the MANOVA Model For Normal Residuals

Both the top and bottom rows display diagnostic plots of the residuals of the Linear MANOVA model with interactions. The graphs in the top row correspond to the model fitted using the untransformed response variables. The bottom row corresponds to the model fitted using the square roots of the response variables. Note that the graphs in the bottom row indicate more normal residuals. Although the lower right graph still shows some heteroscedasticity, it is much less pronounced than in the upper right graph.

## 7. Calculating the Effect of the Factors on the Response Variables

Effects of the factors on the response variables can be calculated independent of the method used to analyze their statistical significance. Hence, both multivariate and univariate methods for calculating these effects give the same results. The results differ only in the standard error of the effects. As can be seen from the diagnostic plots in Figure 6.4, the linear model with interactions, using the square roots of the response variables, results in residuals with less variance than the linear model, with interactions, using the untransformed response variables. Additionally, the residuals plotted against



the fitted values shows non-constant variance in both models. Hence, the constant variance assumption is questionable. However, the ‘Fitted vs. Residuals’ graph of the transformed data shows a much greater degree of homoscedasticity. Thus, we conclude that the model fitted with the square roots of the response variables is the better model. The calculated effects of the various factors on the NmPF parameter values are listed in Table 6.3 and displayed in Figure 6.5.

	NmP	Jumps	SJumps	NmT	Trends	STrends
<b>IF</b>	544982	-5	28	2134200	2	10
<b>FRRT</b>	-2549775	184	-1	-1710384	5	-100
<b>PIFRT</b>	-3206752	1181	-71	-3242005	339	8
<b>RBD</b>	200441	45	-16	-85166	8	-3
<b>RBS</b>	-331458	131	-28	-1460231	-53	-69
<b>AC</b>	-3835864	3120	-463	-6589506	913	-110
<b>FRWDT</b>	-268141	-114	16	-1271939	-31	13
<b>PIFWDT</b>	-3268968	987	-43	-3147961	289	11
<b>RBA</b>	1888221	359	23	1415461	96	9

Table 6.3. Calculated Effects of the Stochastic Parameters.

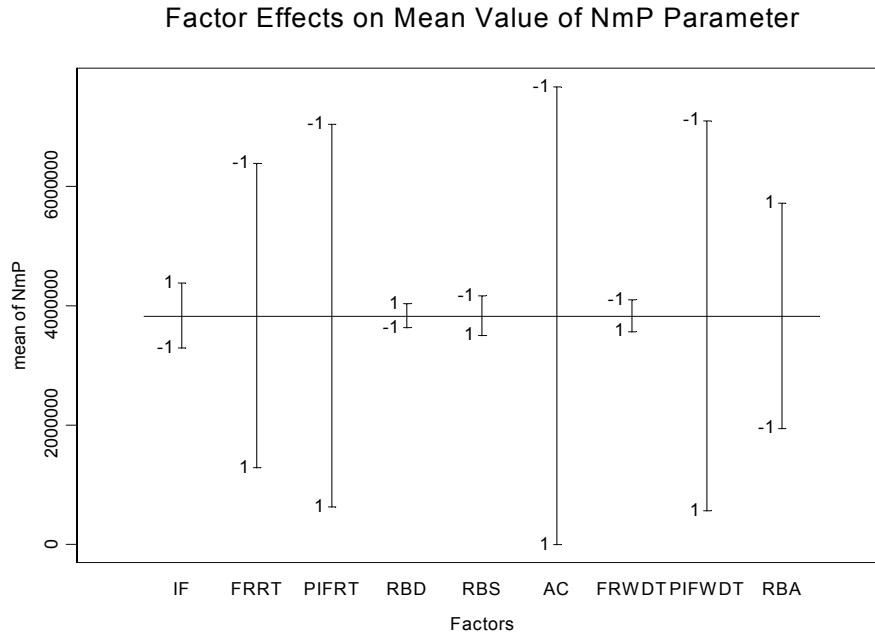


Figure 6.5. Effects of Stochastic Parameters on Mean Value of NmP.

A “1” and “-1” indicate that the parameter value was set to ‘stochastic’ and ‘non-stochastic,’ respectively. Except for IF, RBD and RBA, setting model parameters to ‘stochastic’ reduces the value of NmP.

Each value in Table 6.3 is the effect of the model parameter (row name) on the mean value of the NmPF parameter (column name) when the model parameter is set to ‘stochastic.’ The effects of the factors on the mean value of the NmP parameter can be seen graphically in Figure 6.5. In this figure, the horizontal line running through the center of the graph represents the mean value (3,836,603) of the NmP parameter value for all 64 response surfaces. The vertical lines represent the effect of each factor on the mean value of NmP. For example, in Table 6.3, the effect on the mean value of NmP of setting AC to ‘stochastic’ is –3,835,864. Adding this value to the mean (3,836,603) gives 739. Thus, the estimated effect on the mean value of NmP of making AC stochastic is to reduce it by -3,835,864. Continuing in this manner, it is possible to calculate the cumulative effect of all the factors on the mean value of the NmP parameter. This same procedure can be applied to each of the NmPF parameters in turn. Ideally, we want to set each of the factors at a level that will result in the greatest reduction in non-monotonicity without violating our common sense view of reality.

## 8. A Simple Integer Program to Minimize Non-monotonicity

In order to optimize or find the greatest overall possible reduction in all of the NmPF parameter values—smaller NmP and NmT parameter values imply a reduction in the non-monotonicity of the response surface—the following simple integer program (SIP) is solved:

### Indices

$i$	model parameter {IF, FRRT, ..., RBA}
$j$	NmPF parameter {NmP, ..., STrends}

### Data

$E_{ij}$	The calculated effect of the $i^{\text{th}}$ model parameter on the $j^{\text{th}}$ NmPF parameter value
----------	--

### Integer Decision Variable

$Z_i$	Integer decision variable equal to -1 if the $i^{\text{th}}$ model parameter is ‘not stochastic’ otherwise equal to 1.
-------	--

### Constraints

$Z_i \leq 1$	Force the $i^{\text{th}}$ model parameter to be either stochastic, or
$Z_i \geq -1$	not stochastic
$Z_i$ integer	

### Objective Function

$$\min \sum_{i,j} E_{i,j} Z_i$$

The solution to the SIP is that IF, RBD, and RBA should be set to ‘not stochastic’ and the rest set to ‘stochastic.’ Does this solution make sense? Certainly, the solution is subject to criticism.

Which parameters are made stochastic should pass the ‘common sense’ test. For example, it makes sense to make attrition coefficients stochastic. Attrition during the course of a battle is composed of many inherently random elements, such as detection, engagement, etc. There are periods where attrition is much more intense, and other periods where a lull in the battle results in almost no attrition. Likewise, stochastic decision thresholds make sense. An example of this randomness is the uncertainty a commander faces when trying to decide when to send in his reserves, or, if discretion is the better part of valor, when to withdraw. However, initial force levels are not, normally, uncertain quantities. So, leaving these parameters as deterministic variables makes sense.

Common sense tells us that the time it takes reinforcements to arrive (RBD) ought to be random. The SIP solution is to make this parameter deterministic. Table 6.2 indicates that RBD is statistically significant only in an interaction with a withdrawal threshold. However, its value, relative to the other factors, makes it practically insignificant. Additionally, from Table 6.3, setting this parameter to a ‘stochastic’ level increases the values of NmP, Jumps and Trends, while, at the same time, decreases the values of SJumps, NmT and STrends. There does not appear to be any simple explanation for this rather complicated effect. Thus, whether or not to set this parameter to ‘stochastic’ seems to be a rather subjective call. As we are trying to minimize non-monotonicity, we use the solution of the SIP and set this parameter to its ‘not stochastic’ level. Similar arguments apply to the remaining parameters. There is potential for endless debate between proponents for both sides of such arguments. However, this is missing the point. The real point is that the results of the fractional factorial experiment, conjoined with the knowledge that experience and common sense provide, allow us to decide which parameters should be made stochastic in a logical, rational manner, rather than merely guessing in some haphazard way. Clearly, defensible arguments can be made for several distinct configurations of the model parameters. To illustrate this point, the results of the SIP are applied to the model.

Table 6.4 shows the effects of the model parameters on the NmPF parameter values when the solution of the integer program is applied. Rows highlighted in red correspond to factors set at ‘not stochastic.’ The cumulative estimated effects of the factors on each NmPF parameter value is calculated and listed in the last row of Table 6.4 (highlighted in blue). Even though some might be inclined to use some sort of weighting scheme when calculating the cumulative effects, we use an additive model since the Linear model used to determine which of these factors is significant is an additive model.

	NmP	Jumps	SJumps	NmT	Trends	STrends
<b>IF</b>	-544982	5	-28	-2134200	-2	-10
<b>FRRT</b>	-2549775	184	-1	-1710384	5	-100
<b>PIFRT</b>	-3206752	1181	-71	-3242005	339	8
<b>RBD</b>	-200441	-45	16	85166	-8	3
<b>RBS</b>	-331458	131	-28	-1460231	-53	-69
<b>AC</b>	-3835864	3120	-463	-6589506	913	-110
<b>FRWDT</b>	-268141	-114	16	-1271939	-31	13
<b>PIFWDT</b>	-3268968	987	-43	-3147961	289	11
<b>RBA</b>	-1888221	-359	-23	-1415461	-96	-9
<b>SUM</b>	-16094602	5090	-625	-20886521	1356	-263

Table 6.4. Optimized Effects of Model Parameters on NmPF Parameters.

## 9. Testing the Analysis

To test the analysis, the Dewar model parameters are set according to the levels indicated in Table 6.4. Thirty response surfaces are generated with the model parameters set at these levels. The NmPF parameter values are calculated for each of the resulting surfaces. Then, the column means and standard deviations of the thirty-by-six NmPF parameter matrix are calculated. These statistics are listed in Table 6.5. Figure 6.6 shows the original Dewar model response surface and a response surface chosen randomly from the 30 generated by the empirical test. The NmPF parameter values for the original Dewar model response surface are listed below the figure. Below those values, the mean values from Table 6.5 are listed (highlighted in red).

	Mean	SE
<b>NmP</b>	679.0	30.0
<b>J</b>	13238.4	72.1
<b>SJ</b>	119.4	7.5
<b>NmT</b>	2402.6	37.0
<b>Tr</b>	3984.8	35.0
<b>STr</b>	457.7	10.5

Table 6.5. Mean Values of the NmPF Parameter Values that Result from the SIP Solution

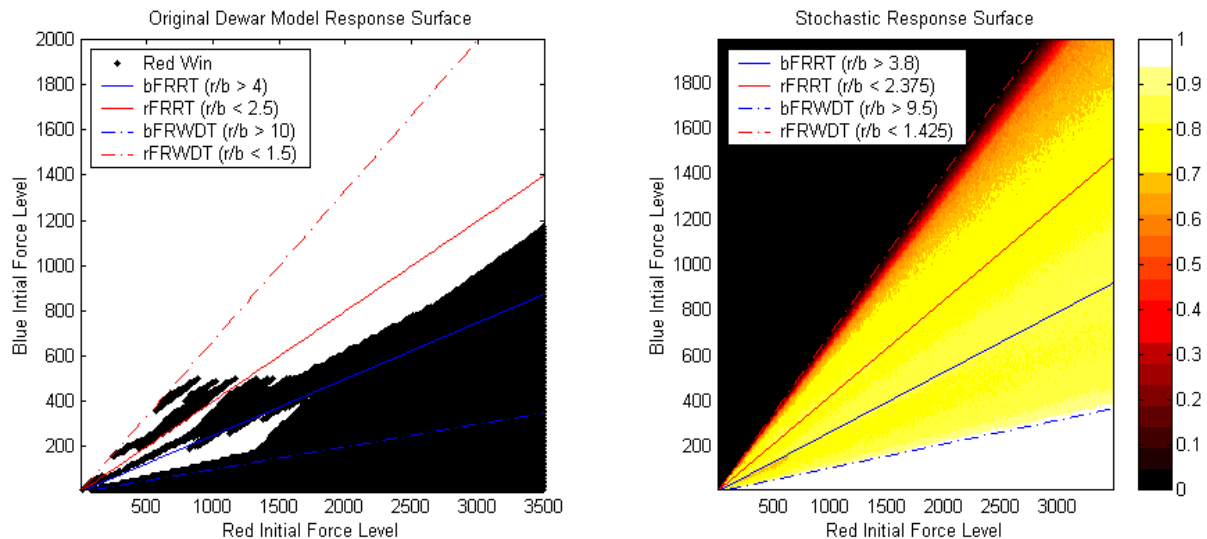


Figure 6.6. Comparing Original Dewar Model With Stochastic Model.

The stochastic response surface (left graph) is more amenable to interpretation than the deterministic response surface (right graph). However, the number of model runs needed to generate a stochastic surface may still be prohibitive for the larger combat models.

The following NmPF parameter values correspond to the left and right graphs, respectively, in Figure 6.6. According to the quantiles in Table 5.2, the graph on the right is ‘monotonic.’

Original Dewar Model Response Surface NmPF Parameter Values					
<b>NmP</b>	<b>Jumps</b>	<b>SJumps</b>	<b>NmT</b>	<b>Trends</b>	<b>STrends</b>
15164400	480	480	8434200	267	267

Optimal Stochastic Model Mean Response Surface NmPF Parameter Values					
<b>NmP</b>	<b>Jumps</b>	<b>SJumps</b>	<b>NmT</b>	<b>Trends</b>	<b>STrends</b>
<b>679</b>	<b>13238</b>	<b>119</b>	<b>2403</b>	<b>3985</b>	<b>458</b>

The left graph in Figure 6.6 is the original Dewar model response surface, and the other is a randomly chosen response surface from the final set of 30 stochastic runs.

Each of these 30 stochastic response surfaces appears identical to the naked eye; the only way to distinguish between them is to calculate the NmPF parameter values. There are four lines superimposed on each graph. These are the force ratio reinforcement (FRRT) and withdrawal thresholds (FRWDT) for each side. In the left graph, the black and white regions represent the initial force levels that result in a Red or Blue win, respectively. This color scheme is reversed in the graph on the right, where the black and white regions correspond to Blue and Red wins, respectively. In the graph on the right, the black and white regions are both bounded by the withdrawal thresholds. The colored region—between the black and white regions in this graph—represents an increasing probability of a Red win, according to the color bar to the right of the graph.

Those combinations of stochastic parameters having the greatest reducing effect on the non-monotonicity of the model also cause the entire region between the bounding thresholds to change. As can be seen in Figure 6.6, a large part of the region that was once clearly dominated by Blue wins is now a region of uncertainty. The same is true for regions previously dominated by Red wins. The right-hand graph in Figure 6.6 appeals to our intuition—i.e., the outcome remains uncertain until one side withdraws or loses the battle. The most significant point to make here is that the stochastic response surface (right graph) is much more amenable to interpretation than the deterministic response surface (left graph).

This is just an example of the way the results of the fractional factorial experiment can be used. As mentioned previously, there are other logically defensible configurations. However, not all configurations are reasonable, and some are better than others. As an example, this is a listing of the factor levels used in the ninth and 54<sup>th</sup> design points of the fractional factorial experiment and their NmPF parameter values:

	<b>IF</b>	<b>FRRT</b>	<b>PIFRT</b>	<b>RBD</b>	<b>RBS</b>	<b>AC</b>	<b>FRWDT</b>	<b>PIFWDT</b>	<b>RBA</b>
9 <sup>th</sup> Design Point	1	-1	-1	1	-1	-1	-1	-1	1
	<b>NmP Jumps SJumps</b>			<b>NmT Trends STrends</b>					
	87020260 1320 1108			103013100 681 663					
	<b>IF</b>	<b>FRRT</b>	<b>PIFRT</b>	<b>RBD</b>	<b>RBS</b>	<b>AC</b>	<b>FRWDT</b>	<b>PIFWDT</b>	<b>RBA</b>
54 <sup>th</sup> Design Point	-1	1	1	-1	1	1	1	1	-1
	<b>NmP Jumps SJumps</b>			<b>NmT Trends STrends</b>					
	719 13182 132			2460 3928 475					

The ninth design point has parameters assigned exactly opposite to the solution of the SIP. The 54<sup>th</sup> design point corresponds, exactly, to the SIP solution. It is obvious that the SIP solution configuration is better than the configuration corresponding to the ninth design point.

Clearly, the use of stochastic parameters requires numerous additional runs to generate a useable stochastic response surface. Remember that each point in a stochastic surface generated in this study is the mean of 1000 replications (see Chapter II for a complete description). The added burden of additional runs may make the use of stochastic parameters, computationally, too expensive for larger combat models. However, when feasible, stochastic modeling can significantly smooth non-monotonicity, making the trend of the response more amenable to interpretation.

## **B. FINAL COMMENTS ON THE RESULTS OF THE FRACTIONAL FACTORIAL EXPERIMENT**

The conclusions it is possible to draw from this chapter are too numerous to list. Because the main goal of this chapter was to determine the effect of stochastic modeling on the Dewar model's response surface, we will limit our conclusions to this topic.

The information obtained from the analysis of the fractional factorial experiment's results, conjoined with experience and common sense, provide a rational basis for deciding which model parameters to make stochastic and permit us to estimate their effects.

As discussed and illustrated above, attrition coefficients have the single greatest effect upon the non-monotonicity of the stochastic response surface of the Dewar model. The next most significant effects come from the reinforcement thresholds, followed by reinforcement block size, and then interactions involving pairs of factors. One of the withdrawal thresholds (FRWDT) does not have a significant effect by itself, but does have a significant effect when interacting with the reinforcement block size (RBS) parameter. It may be useful to unravel some of these interactions, but as that is not the focus of this thesis, we leave it to future study of this fascinating model.

Once the level of the model parameters has been decided—i.e., should the parameter be stochastic or not?—the model can be run and the response surface measured to determine its NmPF values. These values can be used in conjunction with Table 5.2 to classify the stochastic response surface as ‘monotonic’ or not. ‘Monotonic’ surfaces can be interpreted with more confidence than can ‘non-monotonic’ surfaces in terms of what the trend of the surface implies.

Next, we make some concluding remarks and suggest some ideas for further research.



THIS PAGE INTENTIONALLY LEFT BLANK

## VII. CONCLUSIONS

### A. PRIMARY FINDINGS

All three goals of the study were attained, with significant findings in two of them. The three major objectives of this thesis were:

- a. To systematically explore the Dewar model for additional non-monotonic behavior.
- b. To determine the effect of stochastic modeling on the non-monotonic behavior of the Dewar model response surface.
- c. To develop a method for measuring non-monotonicity in the response surface generated by the model.

*The overall goal was to discover if the trend of the response surface could be made more amenable to interpretation without destroying the chaos inherent in the model.*

Latin Hypercube Sampling, an extension of Latin Square sampling into multi-dimensional spaces, was used to thoroughly search the 18 dimensions of the Dewar model. This search discovered non-monotonicity in not only five additional two-dimensional subspaces, but also in one other measure of effectiveness (MOE): ‘length of battle.’ There are 152 other two-dimensional subspaces of the Dewar model that were not searched. It is possible that additional non-monotonic behavior could be found in these subspaces, as well.

Given the pervasiveness of non-monotonicity throughout the Dewar model, we looked at stochastic modeling as a way to smooth the response surface in the presence of the underlying chaotic battle trace. Much of the literature reviewed for this paper indicated that stochastic modeling would be a useful way to deal with non-monotonic behavior of not only the simple combat model studied by Dewar et al., but also other, more complex models. Thus, a fractional factorial design experiment was run to determine the effect of stochastic modeling on the trend of the response surface of the Dewar model. The results were dramatic. Stochastic perturbation can both reduce and

exacerbate the non-monotonic behavior of the response surface. The model parameters having the greatest effect on the reduction of the non-monotonic behavior in the model were the attrition coefficients. When the attrition coefficient parameters were made stochastic, the average reduction in non-monotonicity was approximately three orders of magnitude. It is also important to note that some of the response surfaces generated during the fractional factorial experiment resulted in an increase in non-monotonicity of about two orders of magnitude. In fact, the response surface exhibiting the most extreme non-monotonicity was the design point that had the model parameters set to levels exactly opposite those suggested by the solution to the simple integer program in Chapter VI.

We developed, tested and used the Non-monotonic Parameter Function (NmPF) to measure the reduction in non-monotonicity of the response surface. This function calculates six measures of the non-monotonicity or roughness of the stochastic response surfaces generated by the Dewar model. The function has two user-definable parameters that allow the sensitivity of the function to the response curve to be adjusted. Thus, if the three standard deviation threshold we used in this thesis seems too strict a constraint, it is easily changed.

*Caveat actor et cavendo tutus*; the Dewar model is a relatively small model when compared to the larger, more complex models currently in use throughout DoD. Counting explorations, testing, Latin Hypercube Sampling, fractional factorial experiments, and final empirical testing of analytical results, approximately *ten billion* model runs were made during the course of this study. All of these runs are but an inconsequential fraction of the number of runs it would take to thoroughly explore every corner of the 18-dimensional Dewar model. Clearly, conducting this method of analysis with the large DoD models is impossible. However, the results of this study may suggest useful methods for dealing with non-monotonic output in larger, more complex models.

The bottom line is that non-monotonicity may be more pervasive in combat models than previously suspected. Stochastic modeling can be a viable method for smoothing non-monotonic response surfaces. However, stochastic modeling must be done carefully. When it is, the non-monotonic behavior of the model can be dramatically

reduced, thereby making the trend of the response surface more useful for comparative analyses. Since non-monotonic behavior no longer needs to be ignored or ‘fixed,’ the realism of the model can be retained, while, at the same time, the trends of the response surface can be used to support decisions between competing alternatives. Better models imply better decisions, which, in turn, imply more economical and efficient use of time, money and other resources—and, perhaps, another soldier, sailor, airman or Marine gets to enjoy his pension.

## **B. SUGGESTED FOLLOW-ON RESEARCH**

There are many subtle and not so subtle findings that have not been fully explored in this thesis. The Dewar Model, while relatively small, has some very interesting properties. Its chaotic battle trace and the pervasive non-monotonic behavior of the response surface in many of its subspaces are but two. There is still much information that can be gleaned from this seemingly simple, relatively small dimensional model that may have far-reaching implications for larger, more complex models and simulations. The following list contains recommendations for further study and research:

1. Several subspaces were found to have non-monotonic output with regard to ‘length of battle,’ an often useful measure of effectiveness. The Dewar model executed 200,000 iterations; the stochastic version used in this thesis executed 20,000 iterations. The larger models do not execute nearly this many. What happens to the non-monotonicity of the response surface when very few iterations are executed? If the non-monotonicity increases, larger models may have even worse problems than this study suggests.

2. Using Latin Hypercube Sampling, we searched only the symmetric two-dimensional subspaces of the Dewar model—e.g., Red and Blue initial force level subspace. There are 153 two-dimensional subspaces. We suspect that many of the non-symmetric two-dimensional subspaces also contain significant non-monotonic response surfaces. Why stop at two dimensions? The computer used in this study was a Pentium III, 600 MHz, with 256 Mb of RAM. All told, approximately ten billion model runs were made during this study. Larger, more powerful computers could breeze through that many runs in a day. Three-dimensional and higher studies would be useful in helping

analysts better understand the intricacies of interactions between some of the basic processes in many large combat models. The Dewar model could be considered a type of “distillation” like those being studied in *Project Albert* at the Marine Corps Combat Development Command. (Horne, 2001). It contains some of the same processes present in larger models currently used by DoD, such as Vector In Command (VIC), a U.S. Army combat model. We believe further exploration of the Dewar model will continue to provide useful insights.

3. Tsai and Ellenbogen studied the question of how to find bounds on regions of non-monotonicity. After viewing hundreds of response surfaces in the many two-dimensional subspaces of the Dewar model, we have the impression that, often, the non-monotonicity is not only bounded by decision thresholds, but is also often limited in extent to regions close to these boundaries. Further research in this area would be useful.

4. The main goal of this thesis was to determine the effect of stochastic modeling on the response surface of the Dewar model. Only two random distributions with fixed standard deviations were used. Other distributions could have been used that may have brought to light other pertinent information as to the behavior of the model. A systematic exploration of the effect of various distributions on the response surface would doubtless yield many useful results.

5. Finally, the Dewar model used Lanchester Square Law coupled difference equations to model the attrition process. There are many other attrition calculation mechanisms that could have been used. Do other attrition processes also result in chaotic battle traces and non-monotonic response surfaces?

## APPENDIX A. SPLUS CODE FOR THE NMPF

This appendix lists the SPlus code for the NmPF function. It is usable, as is, in the R language, as well. It is easily translated into C, Java or other computing languages, as it does not rely on any special functionality of SPlus / R. Translating this code into another language will improve the speed of execution.

```
function(x, y, reps, lambda = 6, thresh = 3){
#####
#Usage: NmPF2(x, y, reps, lambda=6, thresh=3)
#
#This function calculates six values that attempt to describe the
#non-monotonicity or roughness of a response curve. It does this by
#determining the overall trend of the response curve using the sign of
#the simple linear regression slope coefficient. Suppose this
#coefficient is positive. Then successive values along the response
#curve that fall below their predecessors are considered non-monotonic
#deviations from the overall trend of the response curve. The
#magnitude of the difference between a non-monotonic deviation and its
#immediate predecessor is calculated. This difference is standardized
#and a zvalue is calculated by the subfunction 'fz'. This zvalue is
#transformed by the subfunction 'fd'.
#####
#Arguments:
#   x      - a vector of x values
#   y      - a vector of response values, each value is assumed to
#           be the mean of a stochastic modeling process. These means
#           are assumed to be normally distributed.
#   reps   - the number of replications used to generate the y values
#   lambda - a tuning parameter that makes the calculated values more
#           or less sensitive to the last argument, 'thresh'
#####
#Return values:
#   NmP     - The total sum of all transformed zvalues.
#   Jumps   - The number of non-monotonic deviations from the overall
#           trend of the response curve.
#   SJumps  - The number of non-monotonic deviations whose magnitude
#           is greater than 'thresh.'
#   NmT     - Two or more successive non-monotonicities is considered
#           a trend. A zvalue for the difference between the
#           extreme values of this trend is calculated and
#           transformed as described above. NmT returns the sum of
#           the transformed values of all these trends.
#   Trends  - The number of trends found by the function.
#   STrends - Number trends with magnitude greater than 'thresh.'
#####
```

```

#'fd' transforms the zvalues, where  $d = z - 3$ .
fd <- function(d, lambda, thresh){
  ifelse(d <= 0, thresh * exp(lambda * d), thresh + d)
}

#'fz' calculates the standardized 'z' value. The formula is based
#on the assumption that the points used in the calculation are the
#means of 'reps' Bernoulli trials. 1e-009 is a fudge factor so
#formula won't explode if the two points are zero and one.
fz <- function(p, reps){
  (abs(p[2]) * sqrt(reps))/
  sqrt(abs(p[1]*(1-p[1])+(p[1]+p[2])*(1-p[1]-p[2]))+1e-009)
}

#Instantiate variables used to count non-monotonicities
z <- NULL
trend <- NULL

#Calculate the slope coefficient of the simple linear
#regression of y on x.
regCoe <- lm(y ~ x)$coe[2]

#The index used to keep track of which point we're looking at.
i <- 1

#A counter to keep track of non-monotonic deviations
jumps <- 0

#While loop is used because we don't know when we'll bump into
#a trend or how long it will last.
while(i < (length(y) - 1)) {

  #'j' is used to keep track of trends
  j <- i

  #If the overall trend is increasing and the next point
  #is non-monotonic
  if((regCoe > 0) & (y[i + 1] - y[i] < 0)) {

    #maxJumpz is biggest non-monotonic jump found
    #thus far along a trend
    maxJumpz <- 0

    #Counts the number of trends found along the response
    #curve
    trendCount <- 0

    #While we're tracking a trend we need a variable to
    #look at successive points
    singleJump <- 0

    #Since the curve is increasing and trend is
    #non-monotonic, need to keep track of
    #monotonic trends. If this reverse trend gets big
    $enough, stop tracking the non-monotonic trend.
  }
}

```

```

revTrendzTotal <- 0

#While loop used since we don't know when the non-
#monotonic trend will stop. We keep tracking the
#trend as long as successive points are non-monotonic
#with respect to the beginning of the non-monotonic
#trend and there are more points on the response curve
#and the sum of monotonic jumps is less than the
#significance threshold, 'thresh.'
while((y[j + 1] - y[i] < 0) & (j < length(y)) &
  (revTrendzTotal < thresh)){

  #Calculate the zvalue of the difference between
  #the start of the trend and the current point.
  multiJumpz <- fz(c(y[i], y[j + 1] - y[i]), reps)

  #Calculate the difference between the current
  #point and the last point.
  singleJump <- y[j + 1] - y[j]

  #If the current point is biggest, save it.
  if(multiJumpz > maxJumpz) {
    maxJumpz <- multiJumpz
  }

  #If the current point is non-monotonic with
  #respect to its immediate predecessor, count it
  #as a non-monotonic jump and calculate its
  #zvalue. Then store the value.
  if(singleJump < 0) {
    jumps <- jumps + 1
    z <- c(z, fz(c(y[j], y[j+1]-y[j]), reps))
  }

  #If the current point is monotonic with respect
  #to its immediate predecessor, then calculate
  #its zvalue and add it to the other
  #monotonic jumps
  if(singleJump > 0) {
    revTrendzTotal <- revTrendzTotal +
      fz(c(y[j], singleJump), reps)
  }

  #Go to the next point.
  j <- j + 1

  #Keep track of how many times trend loop used.

  trendCount <- trendCount + 1
} ### End While Trend Loop ###

```



```

#If the trend loop was used more than once,
#then it was a trend. So save the magnitude of the
#trend.
if(trendCount > 1) {
    trend <- c(trend, maxJumpz)
}

#Start looking for the next non-monotonicity where
#tracking stopped on the last trend.
i <- j

} ### End IF Increasing ###

#This block mirrors the one above, except that the response
#curve is decreasing.
else if((regCoe < 0) & (y[i + 1] - y[i] > 0)) {

    maxJumpz <- 0
    trendCount <- 0
    singleJump <- 0
    revTrendzTotal <- 0

    while((y[j + 1] - y[i] > 0) & (j < length(y)) &
        (revTrendzTotal < thresh)){

        multiJumpz <- fz(c(y[i], y[j + 1] - y[i]), reps)
        singleJump <- y[j + 1] - y[j]

        if(multiJumpz > maxJumpz) {
            maxJumpz <- multiJumpz
        }

        if(singleJump > 0) {
            jumps <- jumps + 1
            z <- c(z, fz(c(y[j], y[j+1]-y[j]), reps))
        }

        if(singleJump < 0) {
            revTrendzTotal <- revTrendzTotal +
                fz(c(y[j], singleJump), reps)
        }

        j <- j + 1
        trendCount <- trendCount + 1
    } ### End While Trend Loop ###

    if(trendCount > 1) {
        trend <- c(trend, maxJumpz)
    }

    i <- j
} ### End IF Decreasing ###

```

```

        #If neither of the conditions is met the point is monotone,
        #put a zero in the z value vector. And go to the next point.
        else {
            z <- c(z, 0)
            i <- i + 1
        }
    } ### End While Loop: Response Points Remaining ###

    #Finished looking at all the points along the response
    #curve. If any zvalues found calculate their 'fd'
    #transformation and save each of these values in the 'nmp' vector.
    if(length(z[z > 0]) > 0) {
        nmp <- apply(as.matrix(z[z>0]- thresh),1,fd,lambda,thresh)
    }

    #If there were no non-monotonicities set 'nmp' equal to zero.
    else {
        nmp <- 0
    }

    #Do the same with the trends found, if any.
    if(length(trend[trend > 0]) > 0) {
        nmt<-apply(as.matrix(trend[trend>0]-thresh),1,fd,lambda,thresh)
    }
    else {
        nmt <- 0
    }

    #Now let the user know what was found.
    result <- signif(c(sum(nmp), jumps, sum(nmp > thresh),
                      sum(nmt), sum(nmt > 0), sum(nmt > thresh)), 3)

    names(result) <- c("NmP", "J", "SJ", "NmT", "T", "ST")

    return(result)
} ### End of Function ###

```

THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX B. NMPF EXPERIMENT RESULTS AND ANALYSIS

Table B.1 is a listing of the full factorial experiment conducted on the NmPF parameter function in Chapter V. For a complete description of the experiment and the analysis of this data, see Chapter V.E –F.

### A. RESULTS OF EXPERIMENT

	fNames	Density	Lambda	SampleSize	NmP	J	SJ	NmT	Trends	STrends
dp 1	1	100	1	100	20.0500	46.48	0.09	16.2400	12.28	0.95
dp 2	1	100	1	500	20.1800	47.58	0.15	15.9100	12.44	0.88
dp 3	1	100	6	100	1.0590	46.98	0.21	4.8810	12.18	1.05
dp 4	1	100	6	500	0.7140	48.10	0.11	3.9090	12.32	0.71
dp 5	1	200	1	100	40.9800	93.90	0.41	32.7500	24.43	2.19
dp 6	1	200	1	500	40.8700	96.79	0.24	31.8700	24.58	1.62
dp 7	1	200	6	100	1.8740	93.05	0.32	10.5600	24.47	2.28
dp 8	1	200	6	500	1.5310	96.82	0.24	8.2070	23.99	1.42
dp 9	2	100	1	100	15.0400	37.38	0.07	10.1600	14.52	0.16
dp 10	2	100	1	500	10.7800	30.95	0.01	5.3370	10.12	0.00
dp 11	2	100	6	100	0.4158	38.06	0.07	0.7651	14.59	0.14
dp 12	2	100	6	500	0.1467	30.90	0.02	0.2873	10.35	0.06
dp 13	2	200	1	100	35.7200	82.70	0.25	26.3400	28.62	0.64
dp 14	2	200	1	500	30.0400	77.43	0.08	19.6000	29.19	0.16
dp 15	2	200	6	100	1.5750	83.17	0.25	4.1240	28.78	0.70
dp 16	2	200	6	500	0.8965	77.76	0.14	1.4840	28.91	0.22
dp 17	3	100	1	100	20.3100	45.83	0.20	19.5200	12.48	2.75
dp 18	3	100	1	500	22.7100	48.65	0.30	25.4900	10.19	3.43
dp 19	3	100	6	100	0.9253	46.92	0.14	11.4000	12.59	2.63
dp 20	3	100	6	500	1.4800	47.35	0.29	20.5300	9.92	3.28
dp 21	3	200	1	100	40.8800	93.40	0.28	35.5000	25.02	3.45
dp 22	3	200	1	500	41.1600	96.53	0.32	39.4700	24.61	5.46
dp 23	3	200	6	100	1.7220	93.73	0.32	14.8800	25.30	3.38
dp 24	3	200	6	500	2.1710	96.76	0.38	25.6100	24.54	5.71
dp 25	4	100	1	100	21.4600	46.74	0.25	25.0800	10.33	3.67
dp 26	4	100	1	500	27.4300	48.66	0.62	40.4100	5.95	3.33
dp 27	4	100	6	100	1.4950	46.96	0.24	20.7100	10.34	3.71
dp 28	4	100	6	500	3.6620	48.71	0.72	39.1300	5.68	3.28
dp 29	4	200	1	100	40.8000	93.60	0.29	39.5400	24.59	5.59
dp 30	4	200	1	500	44.0300	97.16	0.48	50.7500	19.67	6.34
dp 31	4	200	6	100	2.0620	93.11	0.37	24.5700	24.40	5.60
dp 32	4	200	6	500	3.0440	96.97	0.50	42.5100	19.90	6.52
dp 33	5	100	1	100	15.7400	39.34	0.10	11.1900	12.43	0.37
dp 34	5	100	1	500	13.1000	33.98	0.05	8.4310	9.97	0.24
dp 35	5	100	6	100	0.4767	39.18	0.08	2.0970	12.29	0.40
dp 36	5	100	6	500	0.3845	34.32	0.03	1.1310	10.16	0.16
dp 37	5	200	1	100	35.6600	84.81	0.26	26.4900	26.68	1.05
dp 38	5	200	1	500	32.2700	80.48	0.19	21.8300	25.24	0.64
dp 39	5	200	6	100	1.5610	84.81	0.32	5.4520	26.47	1.12
dp 40	5	200	6	500	1.0460	80.89	0.19	3.2240	24.92	0.55

Table B.1. NmPF Full Factorial Experiment Results

THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX C. FRACTIONAL FACTORIAL RESULTS

Table C.1 is a complete listing of the fractional factorial  $2^6$ , resolution V experimental design—the first nine columns of Table C.1—and the results of the experiment—columns ten through 15 of Table C.1. A complete discussion and analysis of this table can be found in Chapter VI.

	IF	FRRT	PIFRT	RBD	RBS	AC	FRWDT	PIFWDT	RBA	NmP	Jumps	SJumps	NmT	Trends	STrends
dp 0	-1	-1	-1	-1	-1	-1	-1	-1	-1	15164400.0000	480	480	8434200.000	267	267
dp 1	1	-1	-1	-1	-1	-1	1	1	1	3017869.0000	6880	1665	13018980.000	2774	1563
dp 2	-1	1	-1	-1	-1	-1	1	1	1	1010321.0000	7880	1007	6010837.000	2637	576
dp 3	1	1	-1	-1	-1	-1	-1	-1	-1	10028930.0000	1822	1681	56021760.000	733	730
dp 4	-1	-1	1	-1	-1	-1	1	1	1	1010533.0000	10444	846	7012324.000	3295	922
dp 5	1	-1	1	-1	-1	-1	-1	-1	-1	2016741.0000	6551	1470	8016766.000	2513	1462
dp 6	-1	1	1	-1	-1	-1	-1	-1	-1	1009322.0000	7694	861	6009837.000	2473	574
dp 7	1	1	1	-1	-1	-1	1	1	1	1008273.0000	10922	859	6009507.000	3275	613
dp 8	-1	-1	-1	1	-1	-1	1	1	-1	1016433.0000	5714	1544	7017475.000	2180	1409
dp 9	1	-1	-1	1	-1	-1	-1	-1	1	87020260.0000	1320	1108	103013100.000	681	663
dp 10	-1	1	-1	1	-1	-1	-1	-1	1	5016461.0000	3666	1606	4016334.000	1140	788
dp 11	1	1	-1	1	-1	-1	1	1	-1	1008988.0000	6778	898	6009677.000	2083	568
dp 12	-1	-1	1	1	-1	-1	-1	-1	1	2012357.0000	8696	1162	8013410.000	3155	1356
dp 13	1	-1	1	1	-1	-1	1	1	-1	1009524.0000	10664	816	7011104.000	3382	989
dp 14	-1	1	1	1	-1	-1	1	1	-1	1007367.0000	11004	801	6008481.000	3375	618
dp 15	1	1	1	1	-1	-1	-1	-1	1	1008341.0000	9389	951	6009317.000	2871	566
dp 16	-1	-1	-1	-1	1	-1	1	-1	1	66014760.0000	2889	1364	45021670.000	904	621
dp 17	1	-1	-1	-1	1	-1	-1	1	-1	1009887.0000	6358	956	7011335.000	1812	806
dp 18	-1	1	-1	-1	1	-1	-1	1	-1	1007628.0000	6575	943	6008410.000	1808	440
dp 19	1	1	-1	-1	1	-1	1	-1	1	9016314.0000	4005	1811	18020350.000	855	632
dp 20	-1	-1	1	-1	1	-1	-1	1	-1	1007739.0000	10808	777	7009046.000	3176	648
dp 21	1	-1	1	-1	1	-1	1	-1	1	2008964.0000	9201	871	8010078.000	2662	719
dp 22	-1	1	1	-1	1	-1	1	-1	1	1006858.0000	9355	866	6008076.000	2623	451
dp 23	1	1	1	-1	1	-1	-1	1	-1	1006764.0000	11123	885	6007815.000	3279	480
dp 24	-1	-1	-1	1	1	-1	-1	1	1	1009495.0000	9147	905	7010982.000	2657	738
dp 25	1	-1	-1	1	1	-1	1	-1	-1	18015820.0000	1733	1184	16015140.000	629	578
dp 26	-1	1	-1	1	1	-1	1	-1	-1	5010516.0000	1870	1049	6011090.000	481	347
dp 27	1	1	-1	1	1	-1	-1	1	1	1007513.0000	9186	946	6008616.000	2624	454
dp 28	-1	-1	1	1	1	-1	1	-1	-1	2009301.0000	6716	905	7011287.000	1913	898
dp 29	1	-1	1	1	1	-1	-1	1	1	1007535.0000	11086	747	7008822.000	3115	561
dp 30	-1	1	1	1	1	-1	-1	1	1	1006569.0000	11314	861	6007959.000	3103	488
dp 31	1	1	1	1	1	-1	1	-1	-1	1007154.0000	6927	896	6007857.000	1986	405
dp 32	-1	-1	-1	-1	-1	1	-1	1	1	758.6826	13459	129	2755.830	4056	529
dp 33	1	-1	-1	-1	-1	1	1	-1	-1	784.5177	13209	136	2798.076	3986	537
dp 34	-1	1	-1	-1	-1	1	1	-1	-1	684.6479	13242	118	2471.311	3996	475
dp 35	1	1	-1	-1	-1	1	-1	1	1	699.8102	13544	121	2398.754	4067	451
dp 36	-1	-1	1	-1	-1	1	1	-1	-1	779.0201	13178	136	2735.437	3985	529
dp 37	1	-1	1	-1	-1	1	-1	1	1	793.2902	13514	138	2774.600	4073	526
dp 38	-1	1	1	-1	-1	1	-1	1	1	723.8387	13571	127	2479.108	4075	468
dp 39	1	1	1	-1	-1	1	1	-1	-1	682.2578	13194	121	2437.831	3988	459
dp 40	-1	-1	-1	1	-1	1	1	-1	1	801.8395	13199	132	2835.400	3978	548
dp 41	1	-1	-1	1	-1	1	-1	1	-1	748.1937	13461	124	2704.865	4105	518
dp 42	-1	1	-1	1	-1	1	-1	1	-1	692.7105	13541	123	2395.594	4105	452
dp 43	1	1	-1	1	-1	1	1	-1	1	707.0958	13265	116	2511.863	4008	474
dp 44	-1	-1	1	1	-1	1	-1	1	-1	801.7554	13511	138	2783.130	4075	527
dp 45	1	-1	1	1	-1	1	1	-1	1	766.0407	13224	126	2732.650	3982	513

Table C.1. Results of Fractional Factorial Experiment

	IF	FRRT	PIFRT	RBD	RBS	AC	FRWDT	PIFWDT	RBA	NmP	Jumps	SJumps	NmT	Trends	STrends
dp 46	-1	1	1	1	-1	1	1	-1	1	671.8752	13283	114	2427.797	3988	461
dp 47	1	1	1	1	-1	1	-1	1	-1	725.5913	13542	128	2506.256	4059	480
dp 48	-1	-1	-1	-1	1	1	1	1	-1	744.4172	13129	124	2702.400	3986	522
dp 49	1	-1	-1	-1	1	1	-1	-1	1	807.8496	13487	140	2851.660	4059	560
dp 50	-1	1	-1	-1	1	1	-1	-1	1	707.7737	13546	121	2516.887	4090	489
dp 51	1	1	-1	-1	1	1	1	1	-1	688.4596	13205	115	2362.304	3982	449
dp 52	-1	-1	1	-1	1	1	-1	-1	1	798.5561	13498	135	2789.250	4045	534
dp 53	1	-1	1	-1	1	1	1	1	-1	788.4349	13101	140	2725.360	3942	521
dp 54	-1	1	1	-1	1	1	1	1	-1	718.8082	13182	132	2460.356	3928	475
dp 55	1	1	1	-1	1	1	-1	-1	1	717.0561	13552	127	2505.151	4058	480
dp 56	-1	-1	-1	1	1	1	-1	-1	-1	758.9594	13569	128	2762.940	4081	522
dp 57	1	-1	-1	1	1	1	1	1	1	784.6353	13213	139	2715.700	3965	524
dp 58	-1	1	-1	1	1	1	1	1	1	692.1646	13279	128	2426.387	3974	470
dp 59	1	1	-1	1	1	1	-1	-1	-1	683.8092	13597	115	2445.109	4102	461
dp 60	-1	-1	1	1	1	1	1	1	1	787.9933	13201	137	2676.150	3971	499
dp 61	1	-1	1	1	1	1	-1	-1	-1	756.4480	13521	124	2689.220	4110	504
dp 62	-1	1	1	1	1	1	-1	-1	-1	661.1131	13561	114	2435.611	4077	459
dp 63	1	1	1	1	1	1	1	1	1	705.5060	13298	121	2463.570	3970	477

Table C.1 (Cont'd). Results of Fractional Factorial Experiment

## LIST OF REFERENCES

- Allen, Gillogly and Dewar, "Non-monotonic Effects in Models with Stochastic Thresholds," *Phalanx*, December 1993.
- Ancker, "Theory of Combat," *Warfare Modeling*, p. 193, Bracken, Kress and Rosenthal Ed., MORS 1995.
- Box, Hunter and Hunter, *Statistics For Experimenters*, John Wiley and Sons, 1978
- Cooper, "Non-Monotonicity and Other Combat Modeling Ailments," *Phalanx*, June 1994.
- Cooper, "Non-monotonicity Revisited," *Phalanx*, September 1994.
- Dewar, Gillogly and Juncosa, "Non-monotonicity, Chaos, and Combat Models," *RAND R-3995-RC*, 1991.
- Gleick, *Chaos: Making a New Science*, Penguin Books, 1987.
- Hausken and Moxnes, "The Micro Foundations of the Lanchester War Equations," *MORS Journal: Special Issue*, vol. 5, no. 3, 2000.
- Huber and Tolk, "Non-monotonicity Effects in Combat Models and Tactical Decision Modeling," *Phalanx*, September 1994.
- Horne, "Beyond Point Estimates: Operational Synthesis and Data Farming," *Maneuver Warfare Science 2001*, Horne and Leonardi Ed.
- Johnson, Isensee and Allison, "A Stochastic Version of the Concepts Evaluation Model (CEM)," *Warfare Modeling*, pp. 87 – 100, Bracken, Kress and Rosenthal Ed., MORS 1995.
- Lauren, "Modeling Combat Using Fractals and the Statistics of Scaling Systems." *MORS Journal: Special Issue*, vol. 5, no. 3, 2000.
- Louer, "More on Non-Linear Effects....," *Phalanx*, March 1993.
- Louer, "Treatment of Non-monotonic Effects in Models for Analyses," *Phalanx*, March 1994.
- Lucas, "How One Randomizes Matters: A Study of Non-monotonicity and Randomness in Combat Analysis," *Phalanx*, March 1997.
- Lucas and Allen, "Non-monotonicity : A Clarification and New Directions," *Phalanx*, September 1997.
- Lucas, "The Stochastic Versus Deterministic Argument for Combat Simulations: Tales of When the Average Won't Do," *MORS Journal: Special Issue*, vol. 5, no. 3, 2000.



- McKay, Beckman and Conover, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, vol. 21, no. 2, May 1979.
- Nagel and Newman, *Godel's Proof*, New York University Press, 1958.
- Palmore, "Dynamical Instability in Combat Models," *Phalanx*, December 1992.
- Palmore, "Research on the Causes of Dynamical Instability in Combat Models," *USSACERL Technical Report 96/95*, August 1996.
- Palmore, "The New Sciences - Chaos, Complexity, and Computability," *MORS Journal: Special Issue*, vol. 5, no. 3, 2000.
- Parnell, Ed., *MORS Journal: Special Issue*, vol. 5, no. 3, 2000.
- Peitgen, Jurgens, and Saupe, *Chaos and Fractals: New Frontiers of Science*, Springer-Verlag, New York, 1992.
- Saeger, Hinch, Dufresne, Dzombar, Hunter, "Instabilities in Complex Deterministic Combat Models," 68th MORS Symposium, USAF Academy, Colorado Springs, Colorado, 20-22 June 2000.
- Saeger, "Understanding Instability in a Complex Deterministic Combat Simulation," not yet published, 24 August 2000.
- S-Plus 2000: Guide to Statistics, Vol 1, Data Analysis Products Division, MathSoft, Seattle, WA, 1999
- Tsai, Ellenbogen, "Bounding Potentially Pathological Nonlinear Behavior in Combat Models and Simulations," *Phalanx*, December 1992.
- Ye, "Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments," *Journal of the American Statistical Association*, vol. 93, no. 444, December 1998.

## INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center  
Fort Belvoir, Virginia
2. Dudley Knox Library  
Naval Postgraduate School  
Monterey, California
3. Marine Corps Representative  
Naval Postgraduate School  
Monterey, California  
[debarber@nps.navy.mil](mailto:debarber@nps.navy.mil)
4. Director, Training and Education, MCCDC, Code C46  
Quantico, Virginia  
[webmaster@tecom.usmc.mil](mailto:webmaster@tecom.usmc.mil)
5. Director, Marine Corps Research Center, MCCDC, Code C40RC  
Quantico, Virginia  
[ramkeyce@tecom.usmc.mil](mailto:ramkeyce@tecom.usmc.mil)  
[strongka@tecom.usmc.mil](mailto:strongka@tecom.usmc.mil)  
[sanftlebenka@tecom.usmc.mil](mailto:sanftlebenka@tecom.usmc.mil)
6. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)  
Camp Pendleton, California  
[doranfv@mctssa.usmc.mil](mailto:doranfv@mctssa.usmc.mil)  
[palanaj@mctssa.usmc.mil](mailto:palanaj@mctssa.usmc.mil)
7. Director, Studies and Analysis Division, MCCDC, Code C45  
Quantico, Virginia  
[thesis@mccdc.usmc.mil](mailto:thesis@mccdc.usmc.mil)
8. Director, Project Albert, MCCDC  
Quantico, Virginia  
[hornege@mccdc.usmc.mil](mailto:hornege@mccdc.usmc.mil)
9. Executive Director, Project Albert, MCCDC  
Quantico, Virginia  
[hornege@mccdc.usmc.mil](mailto:hornege@mccdc.usmc.mil)

10. Dr. Kevin J. Saeger  
OSD/PA&E  
Pentagon, Washington D.C.  
[Kevin.Saeger@OSD.PENTAGON.MIL](mailto:Kevin.Saeger@OSD.PENTAGON.MIL)
11. Dr. Julian Palmore, Department of Mathematics  
University of Illinois at Urbana-Champaign  
1409 W. Green Street  
Urbana, Illinois 61801  
[palmore@uiuc.edu](mailto:palmore@uiuc.edu)
12. Major Juan Ulloa, TRAC  
FT Leavenworth, Kansas  
[mailto:ulloaj@???army.mil](mailto:mailto:ulloaj@???army.mil)
13. Dr. James Dewar, RAND  
1700 Main Street  
P.O. Box 2138  
Santa Monica, California 90407-2138  
[James\\_Dewar@rand.org](mailto:James_Dewar@rand.org)
14. Dr. Thomas Lucas  
Naval Postgraduate School  
Monterey, California  
[twlucas@nps.navy.mil](mailto:twlucas@nps.navy.mil)
15. Dr. Robert Read  
Naval Postgraduate School  
Monterey, California  
[rread@nps.navy.mil](mailto:rread@nps.navy.mil)
16. Dr. Carlos Borges  
Naval Postgraduate School  
Monterey, California  
[borges@nps.navy.mil](mailto:borges@nps.navy.mil)
17. Dr. Paul Sanchez  
Naval Postgraduate School  
Monterey, California  
[pjsanche@nps.navy.mil](mailto:pjsanche@nps.navy.mil)
18. Steve Upton  
Mitre  
[upton@mitre.org](mailto:upton@mitre.org)